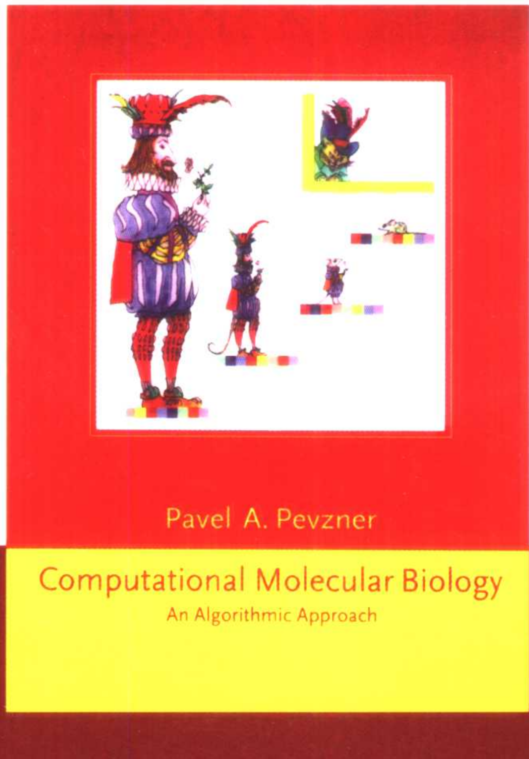
计算分子生物学

——算法逼近

[美] P. A. 帕夫纳　著
王翼飞　等译



Chemical Industry Press
化学工业出版社
现代生物技术与医药科技出版中心

VISIT…

LANZAROTE Caliente.com

计算分子生物学
——算法逼近

［美］P. A. 帕夫纳　著

王翼飞　等译

化学工业出版社
现代生物技术与医药科技出版中心
·北京·

(京) 新登字 039 号

图书在版编目 (CIP) 数据

计算分子生物学——算法逼近/［美］帕夫纳（Pevzner，P. A.）著；王翼飞等译. 北京：化学工业出版社，2004. 8
书名原文：Computational Molecular Biology：An Algorithmic Approach
ISBN 7-5025-5649-4

Ⅰ. 计… Ⅱ. ①帕…②王… Ⅲ. 分子生物学 Ⅳ. Q7

中国版本图书馆 CIP 数据核字（2004）第 061033 号

Computational Molecular Biology：An Algorithmic Approach/by Pavel A. Pevzner
ISBN 0-262-16197-4
Copyright © 2000 by the Massachusetts Institute of Technology. All rights reserved.
Authorized translation from the English language edition published by the Massachusetts Institute of Technology.
本书中文简体字版由 the Massachusetts Institute of Technology 授权化学工业出版社独家出版发行。
未经许可，不得以任何方式复制或抄袭本书的任何部分。
北京市版权局著作权合同登记号：01-2002-5696

计算分子生物学
——算法逼近
［美］P. A. 帕夫纳 著
王翼飞 等译
责任编辑：麻雪丽 莫小曼
责任校对：顾淑云 边 涛
封面设计：关 飞
*
化学工业出版社
现代生物技术与医药科技出版中心 出版发行
（北京市朝阳区惠新里 3 号 邮政编码 100029）
发行电话：（010） 64982530
http：//www.cip.com.cn
*
新华书店北京发行所经销
大厂聚鑫印刷有限责任公司印刷
三河市延风装订厂装订
开本 720mm×1000mm 1/16 印张 $15\frac{3}{4}$ 字数 289 千字
2004 年 3 月第 1 版 2004 年 3 月北京第 1 次印刷
ISBN 7-5025-5649-4/Q · 104
定 价：48.00 元

版权所有 违者必究
该书如有缺页、倒页、脱页者，本社发行部负责退换

中译本序言

拙著《计算分子生物学——算法逼近》的中译本即将出版，对此我非常高兴。近些年来，生物信息学研究在中国很活跃，我知道在中国的许多大学里已制订了生物信息学研究规划。有些大学，如北京大学、清华大学、复旦大学和其他的大学都已积极地投入到许多生物信息学和基因组学的研究行动中。如果没有在中国一流大学中受过教育的有才干的生物信息学家的参与，中国科学家在完成人类基因组的重要部分（人类基因组计划的框架图）研究的成功是不可能的。这种成功的另一个例子是最近的钩端螺旋体全基因组和水稻基因组的测序。如果没有北京大学生物信息中心、中国科学院上海生命科学研究院、中国科学院北京基因组研究所、上海生物信息技术研究中心、香港生物信息中心及其他重要研究机构的努力，生物信息学研究在中国的快速进步是不可能的。最近，在生物信息学和基因组学研究方面，中国已经进入了政府支持和合资经营快速增长的新时期，而且，中国与世界上其他国家在基因组学和生物信息学领域的许多协作研究也已启动。

我希望本书有助于中国新一代生物信息学家的培训，同时我也真诚地希望他们不会感到它是陈旧的。确实，生物信息学的发展如此之快，以致某些领域（在2000年本书刚出版时还是热点）已不再是最前沿的了。反之亦然，某些领域在2000年时甚至并不存在，而今却变成了非常重要的研究课题。但是，本书所展示的一系列算法的核心思想将持续地影响计算分子生物学的研究。我希望本书对研究生和高年级本科生有参考价值。通过阅读本书，使他们不仅可对生物信息学这门崭新的学科有总体上的了解，而且将对他们要从事的研究提供重要的启迪。

最后，我要感谢上海大学的王翼飞教授和他的合作者，他们在将本书翻译成中文的过程中付出了辛勤的劳动。

P. A. 帕夫纳
2004. 3

译者序

历时十年的人类基因组计划不仅产生了海量的生物学数据，在人类探索生命奥秘的征途上迈出了坚实的一步，而且孕育了一门崭新的学科——生物信息学。生物信息学是一门涵盖数学、物理学、化学、生物学和计算机科学的多学科交叉的新兴学科。近年来，生物信息学研究在我国也引起了广泛的重视，许多高等院校和研究机构都开展了这一领域的研究，并开设了相关的研究生课程。虽然相关的书籍也出版了不少，但大多停留于面上的介绍，而对生物信息学研究中用到的数学技术却很少有深入的剖析。其实，计算分子生物学不仅是生物信息学的前身，更是生物信息学的核心部分。可以说，离开了数学技术的支撑，生物信息学的研究是走不了多远的。本书系统而深入地展示了生物信息学研究中涉及的数学技术问题，并对此做了严谨的理论性阐述。

本书作者 P. A. 帕夫纳是美国南加州大学的教授，长期从事计算分子生物学的研究和教学工作，无论在算法的理论方面还是在算法的实现方面都具有丰富的研究与教学经验。本书就是他在宾夕法尼亚州州立大学计算机科学系和南加州大学数学系讲授计算分子生物学课程的基础上形成的，是一本经过教学实践检验的难得的教材。由于现代生物技术和生物信息学的快速发展，原著中介绍的若干内容已不是当前的研究热点，但其中模型的归结方法，特别是包涵的算法的核心思想却仍然不失典型性。因此，本书是从事生物信息学研究的非常出色的入门教材。

要将一本有理论深度的跨学科专著翻译成中文，这不是一项轻松的工作，我们勉为其难。限于译者水平，对原著的理解和翻译难免有不妥甚至错误之处，敬请读者不吝批评指正。如能将译著中的纰漏之处用电子邮件方式（yfwang@mail. shu. edu. cn）通知我们，则将不胜感激。

上海大学数学系生物信息学实验室的研究生顾燕红、倪红春、刘海军、徐东、忻健、陆巍、邵建林、雷耀山、方茜等参加了本书的讨论和翻译，全书最后由王翼飞统稿并校订。

中国科学院上海生命科学院的丁达夫和李亦学研究员、上海大学数学系史定华教授、美国南加州大学汤海旭博士都对本书的翻译给予了热情的支持，并提供了宝贵的意见，原著作者 P. A. 帕夫纳又特意为中译本写了序，在此一并致谢。

本书的翻译工作得到了上海市重点学科建设项目、上海市科委重大项目（02DJ14013）及国家863高技术研究发展计划项目（2002AA234021）的支持。

王翼飞

2004年3月于上海大学

前　言

1985年，我正在俄罗斯的莫斯科找工作，当时我正面临着一个困难的选择。一方面我在一个颇有声望的电子工程研究所获得了一个从事应用组合学研究的职位，另一方面，在莫斯科郊外的俄罗斯生物技术中心NIIGENETIKA，当时正在组建一个计算生物学小组。后一份工作的薪水是前一份的一半，而且连每周的食品包裹“zakaz”也没有，“zakaz”在当时食品匮乏的莫斯科是最重要的工作福利。我至今还不知道电子工程研究所中的人做的是哪种机密研究，因为在我没有签订保密协议之前，他们是不准泄露的。相反，NIIGENETIKA的Andrey Mironov则花了几个小时和我谈了一个崭新的超前的学科——计算分子生物学中的算法问题，于是我选择了NIIGENETIKA。我从没后悔过我的选择，虽然在一段时间里我不得不在莫斯科火车站收集空瓶子来补贴我在NIIGENETIKA的收入，在改革前的莫斯科，收集空瓶子是为数极少的挣外快的合法途径之一。

计算生物学对我而言是一门新的学科，我每个周末都泡在莫斯科列宁图书馆，那是惟一能找到计算生物学论文的地方。当时可用的书仅有Sankoff和Kruskal那本经典的“Time Warps，String Edits and Biomolecules：The Theory and Practice of Sequence Comparison”。由于1985年莫斯科实际上还没有供使用的复印机，我几乎把这本书的每一页都抄在了笔记本上。半年以后我意识到我已经看完了世界上所有的或几乎所有的计算生物学论文。哈，这不是什么了不起的事，这些论文中的一大部分是计算分子生物学的创立者David Sankoff和Michael Waterman写的，而且只有六本期刊需要细看。在接下来的7年里，我每个月去一次图书馆，阅读计算生物学这个领域中发表的所有文献。这样的情形没有持续很久。1992年，我察觉到计算生物学开始蓬勃发展了，因为我第一次没有时间看完所有已发表的计算生物学论文。

由于有些期刊连列宁图书馆也没有收藏，于是我向国外科学家发信索要文章，他们中的许多人好心地寄来了他们的文章预印本。1989年我收到了Michael Waterman寄来的沉重的包裹，里面有一沓即将出版的手稿。其中的一份用公式记述了一个未决问题，而我已经解出了此问题，我没有太担心证明过程就将我的解答寄给了Mike。后来Mike告诉我说，那封信是用非常“俄式的英语”写的，难以看懂，但是他对于有人能够通读他的论文一直到提出未决问题这一情节感到惊讶。不久之后，Mike邀请我去南加州大学和他一起工作，在1992年我讲授了我的第一堂计算生物学课。

本书是我在宾夕法尼亚州州立大学计算机科学系（1992～1995 年）和南加州大学数学系（1996～1999 年）讲授计算分子生物学课程的基础上完成的，它面向计算机科学专业和数学专业的研究生和高年级本科生。对生物信息学感兴趣的分子生物学家也会对本书的部分内容感兴趣。我也希望本书对计算生物学和生物信息学专业人员有所帮助。

本书的基本理论是阐述计算生物学中的算法思想，并说明了它们是如何同分子生物学和生物技术相联系的。为达到这个目的，本书有大量的“不含公式的计算生物学”内容，用简单的方式介绍了生物学动机和计算思想。这些对生物学和计算目标的简化叙述力求使进入这个新领域的计算机科学家和没有充分计算机技术背景的生物学家能够理解本书。例如，在“计算基因搜寻”一章中介绍了许多和搜寻囊肿性纤维化基因有关的计算问题，然后用公式表示了由这些问题导出的组合问题。每一章都有一节引言，叙述计算和生物学思想而无任何公式。本书着重于计算思想而非算法细节，并尽最大努力以简单的方式来介绍这些思想。毫无疑问，达到这一目标的惟一方法就是隐藏一些计算细节和生物学细节，但这在以后可能会被谴责为将计算生物学“卑俗化”。本书的另一个特点就是在每章的最后一节简要地叙述了一些超出本章内容的、重要的、最新的进展。

在计算机科学系开设的计算生物学课程通常先上 2～3 周的“简易分子生物学”导论。我注意到计算机专业的学生（通常对生物学一点儿也不懂）在一开始上生物学导论时如果不联系计算问题，他们的兴趣会很快减弱。如果给生物学家讲述算法而不联系实际的生物学问题，同样的情况也会发生。我发现同时介绍生物学和算法对保持学生在这方面的兴趣非常重要。“计算基因搜寻”一章符合这个目的，虽然这有意简化了生物学和算法的概念。我还发现有些计算生物学家没有清楚地看到计算生物学中不同领域之间的相互联系。例如，从事基因预测的研究人员可能对序列对比算法不甚了解。我尝试着去阐明计算分子生物学中不同领域所涉及的计算思想之间的联系。

本书的内容既包含了计算生物学中新兴的领域，也包含了其中一些相当老的领域。例如，“计算蛋白质组学”、“序列比较”和“DNA 阵列”、“基因组重排”几章中的大部分内容还从未在其他的书中发表过。同时也有一些主题，比如在“限制酶切作图”一章中提到的是相当老的，其中所叙述的实验方法现在已经很少使用了。将这些相当老的计算思想包含在内的原因有两个：一是向新入门的人展示该领域中这些思想的来龙去脉，同时提醒他们计算生物学中的热点转移很快；二是这些计算思想常常在不同的应用领域里获得第二次生命。例如，限制酶切作图中那些几乎被遗忘的技术在计算蛋白质组学这个研究热点中得到新生。还有许多诸如此类的例子（比如一些和杂交测序有关的思想现在正用在大规模鸟枪法拼接上），所以我觉得同时包含老的和新的计算方法是重要的。

简要地谈一下本书中有关应用部分与理论部分间的孰轻孰重。毫无疑问，21世纪的生物学家必须要知道离散数学和算法的基本原理——至少能用公式来表示他们研究中导出的算法问题。在计算生物学中，将生物学问题恰当地公式化也许是研究中最困难的部分，至少和解这些问题同样困难。那么，我们如何教学生用计算术语来公式化地表示生物学问题呢？因为我不知道，所以我用一个故事代替。

20 年前我从大学毕业，在莫斯科登了一则“数学咨询”的广告。我的客户主要是各个不同领域中的 Cand. Sci. （俄语，类似于 Ph. D.）学员，他们缺少良好的数学背景，为毕业证书（或至少是数学这部分）而希望得到帮助。我接触到从“机场清雪设备盘存的最优化”到“向特许经销商交付汽车的时间安排”大量不同领域的课题。在所有的项目中，最难的部分是断定这是一个怎样的计算问题并使其公式化，接下来的求解就是直接利用已有的方法了。

我永远忘不了一位来访者，他 40 来岁，彬彬有礼，体格健壮。他和别的客户不同，没有描述他的研究领域，而是让我解一个微分方程。一开始我很高兴，可结果却发现那个方程是没有意义的。要想出如何解决这个问题的惟一办法就是回到原始的应用问题并推导出一个新的方程。这位来访者对此有点踌躇，不过这是他获得 Cand. Sci. 学位的惟一办法，于是他开始透露一点他研究领域里的细节。到了最后，我领会到他研究的是让某个物体着陆在一个摇晃的平台上。这也使我明白了他为何从不给我他的电话号码，他是一位从事机密研究的军官，摇晃的平台是船，着陆的物体是飞机。我希望 20 年后揭露这个故事不会伤害他的军事生涯。

关于生物问题的公式化，大自然泄露的比那位军官还要少。此外，一些生物问题如果用足够多的公式来表示，则会变得太花哨，有时会遮蔽和掩盖计算思想。由于本书着重于计算思想而非技术细节，我有意用简化的公式来清楚地表达计算思想。这会使人感到本书太理论化，但我想不出别的方法来教授生物学中的计算思想。换句话说，要在真实的船上着陆真实的飞机之前，学生必须学会如何在玩具船上着陆玩具飞机。

我要强调一下，本书不打算一成不变地探讨计算生物学中的所有领域。当然，本书所选择的主题受我的爱好和研究兴趣的影响。计算生物学中的一些重大领域——其中大部分是值得注意的——没有在书中涉及，如 DNA 统计、遗传作图、分子进化、蛋白质结构预测和功能基因组。这些领域中的每一个都可以单独写成一本书，其中有些已经完成了。例如，Waterman 在 1995 年写的书中[357]极好地介绍了 DNA 统计，Gusfield 在 1997 年写的书中[145]详细地解释了字符串算法，Salzberg 等在 1998 年写的书中[296]有几章全面地描述了蛋白质结构预测。Durbin 等在 1998 年写的书[93]和 Baldi 与 Brunak 在 1997 年写的书[24]是着重于隐

马氏模型和机器学习的专著。Baxevanis 与 Ouellette 在 1998 年写的书[28]是一本优秀的生物信息学实用指南，那本书更注重算法的应用而非算法本身。

我要感谢许多曾经教授给我计算分子生物学各方面知识的人。Andrey Mironov 教导我的常识也许是任何应用研究中最重要的部分。当我从莫斯科来到洛杉矶的时候，Mike Waterman 是我科学研究和生活上的最好的导师。尤其是他耐心地教导我，每一篇准备发表的论文都必须至少反复查看十几遍。虽然这个规矩使本书的出版延迟了几年，但我仍将它认真地传授给我的学生。我以前的学生 Vineet Bafna 和 Sridhar Hannenhalli 热心地教我他们所了解的知识，并加入了我这个艰难、长期的项目。我还要感谢 Alexander Karzanov 教我组合最优化，其中一些思想在我的计算生物学研究中非常有用。

我要感谢我的同事们和合著者：Mark Borodovsky，我和他共同研究过 DNA 统计，1985 年他使我相信计算生物学有一个远大的前景；Earl Hubbell，Rob Lipshutz，Yuri Lysov，Andrey Mirzabekov 和 Steve Skiena 是我做 DNA 阵列研究时的同事；Eugene Koonin，在第一个细菌基因组测序完成前，我就和他试着分析完整的基因组；Norm Arnheim，Mikhail Gelfand，Melissa Moore，Mikhail Roytberg 和 Sing-Hoi Sze 是我做基因搜寻时的同事；Karl Clauser，Vlado Dancik，Maxim Frank-Kamenetsky，Zufar Mulyukov 和 Chris Tang 是我做计算蛋白质组学时的同事；Eugene Lawler，Xiaoqiu Huang，Webb Miller，Anaoly Vershik 和 Martin Vingron 是我做序列比较时的同事。

我还要感谢许多同事，我和他们一起讨论计算分子生物学的各个方面，这都直接或间接地影响到本书：Ruben Abagyan，Nick Alexandrov，Stephen Altschul，Alberto Apostolico，Richard Arratia，Ricardo Baeza-Yates，Gary Benson，Piotr Berman，Charles Cantor，Radomir Crkvenjakov，Kun-Mao Chao，Neal Copeland，Andreas Dress，Radoje Drmanac，Mike Fellows，Jim Fickett，Alexei Finkelstein，Steve Fodor，Alan Frieze，Dmitry Frishman，Israel Gelfand，Raffaele Giancarlo，Larry Goldstein，Andy Grigoriev，Dan Gusfield，David Haussler，Sorin Istrail，Tao Jiang，Sampath Kannan，Samuel Karlin，Dick Karp，John Kececioglu，Alex Kister，George Komatsoulis，Andrzey Konopka，Jenny Kotleman，Leonid Kruglyak，Jens Lagergren，Gadi Landau，Eric Lander，Gene Myers，Giri Narasimhan，Ravi Ravi，Mireille Regnier，Gesine Reinert，Isidore Rigoutsos，Mikhail Roytberg，Anatoly Rubinov，Andrey Rzhetsky，Chris Sander，David Sankoff，Alejandro Schaffer，David Searls，Ron Shamir，Andrey Shevchenko，Temple Smith，Mike Steel，Lubert Stryer，Elizabeth Sweedyk，Haixi Tang，Simon Tavarè，Ed Trifonov，Tandy Warnow，Haim Wolfson，Jim Vath，Shibu Yooseph 等。

我和麻省理工学院出版社的 Bob Prior 和 Michael Rutter 合作很愉快。我要感谢 Amy Yeager 编辑了本书，Mikhail Mayofis 设计了封面，Oksana Khleborodova 举例说明了基因预测的算法步骤。我还要感谢那些曾经支持过我研究的部门：能源部、美国国立卫生研究院和美国国家科学基金会。

最后，很感谢 Paulina 和 Arkasha Pevzner 好心地保持安静并容忍我在写书时的心不在焉。

编　者

内 容 提 要

计算分子生物学是数学、计算机科学和生物科学交叉的新兴领域之一。本书覆盖了算法和组合问题的主要领域，并揭示了这些问题是如何同分子生物学和生物技术相联系的。本书包含一个内容丰富的“无公式的计算分子生物学”部分，用相对简单的方式介绍了生物学概念和计算思想，但涉及的内容却是许多基因组基本问题及其结果的核心思想。

本书力求用通俗的写法表达如何用深刻的数学思想解释复杂的生物难题，这使得没有经过生物学训练的计算机科学家和只有有限的计算机科学背景的生物学家都能够理解本书。

本书适合作为数学、计算机科学、生物信息学、生物物理学等交叉学科的教材，也可作为相关专业研究人员的参考书。

目　　录

1　计算基因搜寻 …… 1

1.1　引言 …… 1

1.2　遗传作图 …… 1

1.3　物理作图 …… 4

1.4　测序 …… 6

1.5　相似性搜寻 …… 8

1.6　基因预测 …… 9

1.7　突变分析 …… 10

1.8　比较基因组 …… 11

1.9　蛋白质组学 …… 13

2　限制酶切作图 …… 15

2.1　引言 …… 15

2.2　双酶切问题 …… 16

2.3　双酶切问题的多重解 …… 17

2.4　着色图中的交错圈 …… 20

2.5　交错欧拉圈的变换 …… 21

2.6　物理图谱与交错欧拉圈 …… 23

2.7　部分酶切问题 …… 25

2.8　同效集 …… 26

2.9　若干其他的问题和方法 …… 28

2.9.1　光学作图 …… 28

2.9.2　加探针部分酶切作图 …… 29

3　图谱装配 …… 30

3.1　引言 …… 30

3.2　非惟一探针的作图 …… 33

3.3　惟一探针的作图 …… 36

3.4　区间图 …… 37

3.5　用限制酶切片段指纹图谱作图 …… 39

3.6 若干其他的问题和方法…………………………………………………… 41
3.6.1 Lander-Waterman 统计…………………………………………… 41
3.6.2 筛选克隆文库………………………………………………………… 41
3.6.3 放射杂交作图………………………………………………………… 42

4 测序…………………………………………………………………… 44
4.1 引言……………………………………………………………………… 44
4.2 重叠、排列和共有序列……………………………………………… 45
4.3 双管鸟枪测序………………………………………………………… 46
4.4 若干其他的问题和方法……………………………………………… 47
4.4.1 最短超字符串问题…………………………………………………… 47
4.4.2 DNA 测序的结束阶段 ……………………………………………… 48

5 DNA 阵列……………………………………………………………… 49
5.1 引言……………………………………………………………………… 49
5.2 杂交测序………………………………………………………………… 51
5.3 SBH 和最短超字符串问题 ………………………………………… 52
5.4 SBH 和欧拉路问题…………………………………………………… 53
5.5 惟一序列重构的概率………………………………………………… 56
5.6 字符串重排……………………………………………………………… 57
5.7 2-最优欧拉圈 ………………………………………………………… 60
5.8 杂交法定位测序……………………………………………………… 62
5.9 DNA 阵列的设计 …………………………………………………… 62
5.10 DNA 阵列的分辨率………………………………………………… 64
5.11 多探针阵列与均匀阵列的比较 …………………………………… 65
5.12 DNA 阵列的制造…………………………………………………… 66
5.13 若干其他问题和方法 ……………………………………………… 69
5.13.1 利用通用碱基的 SBH …………………………………………… 69
5.13.2 自适应 SBH ……………………………………………………… 70
5.13.3 SBH-类型的鸟枪测序法………………………………………… 70
5.13.4 DNA 阵列的保真探针…………………………………………… 70

6 序列比较……………………………………………………………… 71
6.1 引言……………………………………………………………………… 71
6.2 最长共同子序列问题………………………………………………… 73

6.3 序列联配 …… 75
6.4 局部序列联配 …… 76
6.5 有缺口罚分的联配 …… 77
6.6 空间效率高的序列联配 …… 77
6.7 杨氏表 …… 79
6.8 最长共同子序列的平均长度 …… 82
6.9 广义序列联配和对偶性 …… 84
6.10 序列比较的原始对偶方法 …… 86
6.11 序列联配和整数规划 …… 87
6.12 近似字符串匹配 …… 88
6.13 依靠数据库的序列比较 …… 89
6.14 多重过滤 …… 90
6.15 若干其他的问题和方法 …… 92
6.15.1 参数的序列联配 …… 92
6.15.2 联配统计量和相变 …… 92
6.15.3 次优序列联配 …… 93
6.15.4 串联重复片段的联配 …… 93
6.15.5 筛选数据库中的搜索结果 …… 93
6.15.6 文本之间的统计距离 …… 93
6.15.7 RNA 折叠 …… 94

7 多重联配 …… 95
7.1 引言 …… 95
7.2 计算多重联配的得分 …… 96
7.3 装配成对联配 …… 97
7.4 多重联配的近似算法 …… 98
7.5 装配 l-路联配 …… 99
7.6 点阵和图像重构 …… 100
7.7 点阵相乘的多重联配 …… 101
7.8 若干其他的问题和方法 …… 102
7.8.1 借助于进化树的多重联配 …… 102
7.8.2 在编辑图中切割角 …… 102

8 发现 DNA 中的信号 …… 103
8.1 引言 …… 103

8.2 Edgar Allan Poe 和 DNA 语言学 …… 104
8.3 适合傻子的最好的赌博 …… 106
8.4 Conway 等式 …… 107
8.5 DNA 中的频繁词 …… 108
8.6 共有词的分析 …… 110
8.7 CG 岛和“公平赌场” …… 111
8.8 隐马氏模型 …… 112
8.9 Elkhorn 赌场和隐马氏模型参数估计 …… 114
8.10 剖面隐马氏模型联配 …… 114
8.11 吉布斯抽样 …… 116
8.12 若干其他的问题和方法 …… 116
8.12.1 找出带缺口的信号 …… 116
8.12.2 在带偏倚频率样本中找出信号 …… 116
8.12.3 信号寻找中字母表的选择 …… 117

9 基因预测 …… 118
9.1 引言 …… 118
9.2 基因预测的统计学方法 …… 119
9.3 基于相似性的基因预测方法 …… 121
9.4 剪接联配 …… 123
9.5 反向基因搜索和 cDNA 中外显子定位 …… 126
9.6 关于基因的多次提问策略 …… 127
9.7 可变剪接与癌症 …… 128
9.8 若干其他的问题和方法 …… 130
9.8.1 基因预测的隐马氏模型 …… 130
9.8.2 细菌基因预测 …… 131

10 基因组重排 …… 132
10.1 引言 …… 132
10.2 断点图 …… 141
10.3 难以排序的排列 …… 142
10.4 期望的反序距离 …… 143
10.5 有符号排列 …… 145
10.6 交叉图和障碍 …… 146
10.7 排列的等价变换 …… 149

10.8 搜索安全反序…… 153
10.9 清除障碍…… 156
10.10 反序距离的对偶定理 …… 160
10.11 反序排序算法 …… 163
10.12 人基因组变换成老鼠基因组 …… 164
10.13 加帽染色体 …… 168
10.14 帽子和尾部 …… 171
10.15 基因组距离的对偶定理 …… 173
10.16 基因组重复 …… 175
10.17 若干其他的问题和方法 …… 175
10.17.1 基因组重排和系统发育研究…… 175
10.17.2 依照反序排序的快速算法…… 176

11 计算蛋白质组学…… 177
11.1 引言…… 177
11.2 肽测序问题…… 178
11.3 谱图…… 180
11.4 学习离子类型…… 182
11.5 给谱图中的路径打分…… 183
11.6 肽测序与反对称路径…… 185
11.7 肽鉴定问题…… 186
11.8 谱的卷积…… 186
11.9 谱联配…… 188
11.10 依靠谱联配肽 …… 191
11.11 若干其他的问题和方法 …… 192
11.11.1 从蛋白质组学到基因组学…… 192
11.11.2 大尺度蛋白质分析…… 192

12 问题…… 193
12.1 引言…… 193
12.2 限制酶切作图…… 193
12.3 图谱装配…… 195
12.4 测序…… 196
12.5 DNA 阵列 …… 197
12.6 序列比较…… 199

12.7 多重联配 …… 202
12.8 探测 DNA 中的信号 …… 203
12.9 基因预测 …… 203
12.10 基因组重排 …… 204
12.11 计算蛋白质组学 …… 206

13 必须了解的分子生物学 …… 208

参考文献 …… 211

索引 …… 230

1 计算基因搜寻

1.1 引言

囊肿性纤维化是一种伴随着反复出现的呼吸感染和异常分泌物的致命疾病。每 2500 个孩子中就有一个被诊断患有这种疾病。每 25 个高加索人中就有一个携带着有缺陷的囊肿性纤维化基因，一个孩子若从父母遗传而得到这种有缺陷的基因就将得病。

20 世纪 80 年代中期，生物学家对引起囊肿性纤维化的基因了解甚少，还没有可靠的胎儿期诊断方法。当时，对许多遗传疾病治疗的最好的期望仍为找到有缺陷的基因。对于囊肿性纤维化（CF）基因的寻找开始于 20 世纪 80 年代早期，在 1985 年，三组科学家同时独立地证明了 CF 基因位于第 7 号染色体上。到了 1989 年，搜索已被集中在第 7 号染色体上很狭窄的区域，并终于找到了 1480 氨基酸长度的 CF 基因。这一发现导致了有效的医学诊断技术和关于 CF 的潜在的治疗希望。在 20 世纪 80 年代后期，囊肿性纤维化基因的搜寻是一项艰苦的工作。自从发现了数以千计的医学上重要的基因后，其他基因的寻找就普遍地启动了。基因搜寻涉及许多计算问题，下面将对此做一些简要的回顾。

1.2 遗传作图

就像制图师绘制古代世界的地图一样，生物学家在过去的 30 余年中费力地绘制人类的 DNA 图谱。其目的是确定基因和其他标记物在各条染色体上的位置，了解基因组的蓝图。

当寻找 CF 基因起步的时候，科学家们对该基因的本性或它在基因组中的定位没什么线索。基因搜寻通常开始于遗传作图，它可以提供基因在人类染色体上的近似位置（通常在几百万个核苷酸长度内）。为了理解遗传作图中的计算问题，我们利用一个极其简化的单染色体机器人的遗传作图的模型。每个机器人有 n 个基因（其次序未知），每个基因仅有两种状态 0 或者是 1，这导致了两种表现型（物理特性）：红色和褐色。如果假定 $n=3$，且机器人的三个基因决定它头发、眼睛和嘴唇的颜色，那么 000 是全红的机器人（红头发、红眼睛和红嘴唇），而 111 则为全褐色的机器人。虽然我们可以观察机器人的表现型（即它们的头发、眼睛和嘴唇的颜色），但并不知道这些基因在其基因组中的次序。幸运的是，机器人可能有孩子，这可以帮助我们来绘制机器人的遗传图谱。

机器人 $m_1 \cdots m_n$ 和 $f_1 \cdots f_n$ 的孩子或是机器人 $m_1 \cdots m_i f_{i+1} \cdots f_n$ 或是机器人 $f_1 \cdots f_i m_{i+1} \cdots m_n$，对于任一基因重组位置 i，其中 $0 \leqslant i \leqslant n$。每对机器人可有 $2(n+1)$ 种不同的孩子（可能其中一些是相同的），而在位置 i 重组的概率是 $\frac{1}{n+1}$。

遗传作图问题 给出全红和全褐机器人的一大群孩子的表现型，判定机器人的基因次序。

分析不同表现型对的出现频率可以推断出基因的次序。计算一个全红和一个全褐机器人的孩子有不同颜色的头发和眼睛的概率 p。如果头发基因和眼睛基因在基因组上是连续的，那么这些基因间重组的概率是 $\frac{1}{n+1}$。如果头发基因和眼睛基因不连续，那么孩子有不同的颜色的头发和眼睛的概率是 $p=\frac{i}{n+1}$，其中 i 是这些基因在基因组中的距离。测算孩子们的 p 可以帮助我们估计基因间的距离，从而发现基因次序，并重构遗传图谱。

在机器人的世界，孩子的染色体包含两个片段：一个片段来自机器人母亲而另一个片段来自机器人父亲。在更准确的（但仍然是不现实的）重组模型中，孩子的基因组被定义为父亲和母亲基因组的任意个片段的嵌合体，例如 $m_1 \cdots m_i f_{i+1} \cdots f_j m_{j+1} \cdots m_k f_{k-1} \cdots$。在这种情况下，在两个基因间重组的概率与这两个基因间的距离成比例，正如前述，基因分离得越远，在两者间重组的可能性越大。如果两个基因彼此非常靠近，那么它们重组的可能性就很小。因此，在全红和全褐机器人的孩子们中，相邻基因意味着出现相同的表现型（全红或全褐）频率更高，所以，生物学家可以通过表现型的成对地出现频率来推断基因的次序。据此，Sturtevant 在 1913 年为果蝇中六个基因建立了第一个遗传图谱。

尽管人类遗传学比机器人遗传学更复杂，但这个简单的机器人模型却捕捉到了许多在后面遗传作图算法中提出的算法思想。一个复杂问题是人类的基因是成对出现的（更不要提它们是分布于 23 对染色体中）。在每对基因中，一个基因从父亲那里继承而另一个则从母亲那里继承。因此，人类基因组可包含一个基因，其状态为 1（红眼睛），位于一个染色体上；而一个状态为 0 的基因（褐色眼睛）则位于同一对的另一个染色体上。如果 $F_1 \cdots F_n | \mathscr{F}_1 \cdots \mathscr{F}_n$ 代表父亲基因组（每个基因被表示为两个拷贝 F_i 和 $\mathscr{F}_i$）而 $M_1 \cdots M_n | \mathscr{M}_1 \cdots \mathscr{M}_n$ 代表母亲的基因组，于是孩子的基因组可以表示为 $f_1 \cdots f_n | m_1 \cdots m_n$，其中 f_i 相当于 F_i 或 $\mathscr{F}_i$ 中的任何一个，而 m_i 相当于 M_i 或 $\mathscr{M}_i$ 中的任何一个。例如，父亲 11 | 00 和母亲00 | 00可能有 4 个不同的孩子：11 | 00（无重组）、10 | 00（重组）、01 | 00（重组）和 00 | 00（无重组）。绘制人类和机器人遗传图谱的基本思想是相似的，因为在邻近基因间的重组是很稀少的，在孩子们中基因重组的比例给出了在染色体上基因

之间距离的指标。

另一个复杂问题是基因型的差别并不一定导致表现型的差别。例如，人类有一个基因叫 ABO 血型，在人群中血型有三种状态——A 型、B 型和 O 型。该基因存在六种可能的表现型——AA 型、AB 型、AO 型、BB 型、BO 型和 OO 型，但实际只表现出四种。这种情况使研究人员无法明确地由表现型推断出基因型。由此可见，眼睛的颜色或血型对于建造遗传图谱并不是最重要的事物。生物学家们建议在遗传图谱绘制中用遗传标记作为基因的适宜的替代物。为了定位一个新基因，必须有大量已经定位的标记，它们均匀地散布在染色体上。

定位机器人的基因的能力是基于不同的机器人具有表现型的可变性。例如，如果所有的机器人都有褐色的眼睛，那么这眼睛基因就有可能无法定位。在人类基因组中有许多变异，它们并不直接表达为表现型。例如，有一半的人在基因组的特定位置上有一个核苷酸 A，而另一半人的相同位置上是核苷酸 T，这对遗传图谱来说是一个非常好的标记。这样的突变会在任何基因的外部发生，并且根本不会影响表现型。Botstein 等[44]建议使用这种可变位置作为作图的遗传标记。因为在基因组的给定位置上标出字母是无法用实验来实现的，他们建议使用一种称为限制性片段长度多态性（restriction fragment length polymorphism，RFLP）的技术来研究可变性。

Hamilton Smith 在 1970 年发现一种限制酶 HindⅡ能在序列中出现 GTGCAC 或 GTTAAC（限制酶切位点）的每个地方切断 DNA 分子。在 RFLP 分析中，人类的 DNA 被像 HindⅡ的限制酶在每个限制位置上切断成百万条限制片段，每个大概是几千个核苷酸。然而任何影响限制位置（如 HindⅡ的 GTGCAC 或 GTTAAC）的突变会使得某个切断失败并且使两个应该切断的限制片段 A 和 B 合并成一个片段 $A+B$。这个 RFLP 分析的关键就是检测限制片段在长度上的变化。

凝胶电泳分开了限制片段，并且一个标记了的 DNA 探针被用于测定和探针杂交的限制片段的大小。在不同个体中的这些限制片段的长度可变性可作为遗传标记，因为单个核苷酸的突变可能会丢失（或新建）一个限制酶的位置并改变相应片段的长度。例如，如果一个标记了的 DNA 探针和片段 A 杂交，并分割片段 A 和 B 的限制位置因为突变而丢失，那么探针检测 $A+B$ 来替代 A。Kan 和 Dozy[183]找到一种通过鉴别在镰状细胞贫血基因附近的 RFLP 标记来诊断镰状细胞贫血症的新方法。

RFLP 分析将遗传图谱的绘制变成高度竞争的比赛，而且在短时间内成功地找到了导致亨廷顿疾病[143]、杜兴肌营养不良[81]和眼癌[60]的基因。在里程碑似的出版物中，Donis-Keller 等[88]建立了第一个人类基因组的 RFLP 图谱，大约每隔一千万个核苷酸定位一个 RFLP 标记。在此研究中，393 个随机探针被用来研

究三代 21 个家庭的 RFLP。最后，重组的计算分析确定了染色体上的 RFLP 标记的排序。

在 1985 年，重组研究使得对囊肿性纤维化基因的搜索已缩小到了第 7 号染色体中在标记了 *met*（一个与癌有关的基因）和 D7S8（RFLP 标记）之间的区域。这个区域的长度大概是一百万个核苷酸，且在囊肿性纤维化基因被发现之前会随着时间的消逝而进一步缩小。在遗传作图后出现的物理作图更加缩小了搜索范围。

1.3 物理作图

物理作图可按照下述类似的方式来理解，设想一本书的几个拷贝被剪刀剪切成了成千上万块碎片。每份拷贝是由独立的方法来剪的，所以每份拷贝的碎片有可能和其他拷贝的碎片重叠。对每块碎片和一栏关键词中的每个词语，要断定碎片是否包含关键词。对于给出的资料，希望决定碎片重叠的模式。

这个过程开始于将 DNA 分子切成小片段（例如用限制酶），在 CF 工程中 DNA 被切割成约 50kb 长短的片段。为了研究每个单独的片段，生物学家需要得到它们的许多拷贝。这可通过克隆这些片段来实现。克隆化是把 DNA 片段插入一些可自我复制的宿主中。自我复制过程于是产生了大量的片段拷贝，从而使它的结构可以被研究。一个片段以这种方式复制被称为克隆。

结果是生物学家们得到了一个克隆库，它包含了成千上万个来自相同的 DNA 分子的克隆（每个代表了一个短的 DNA 片段）。克隆库中的克隆可能出现重叠（这可通过利用不同的限制酶来切割 DNA 以产生重叠的酶切片段而实现）。在克隆库建成后，生物学家希望排列克隆，也就是说要重构克隆在 DNA 分子上的相对位置。在克隆库的构建过程中信息可能丢失，因此重构过程由克隆的“指纹”识别开始。其思想是利用一个容易确定的“指纹”来描述每个克隆，这可以把克隆看成一套“关键词”。如果两个克隆基本重叠，那么它们的指纹应该是相似的。如果不重叠的克隆不可能有相同的指纹，那么这些指纹可以让生物学家用于区分重叠和非重叠的克隆，并且可用于重构克隆的次序（物理图谱）。克隆的限制片段的大小或一系列与克隆杂交的探针提供了这样的指纹。

为了绘制囊肿性纤维化基因的图谱，生物学家使用称为染色体步查（chromosome walking）和染色体跳查（chromosome jumping）的物理作图技术。考虑到 CF 基因已被连接到 RFLP D7S8，与这种 RFLP 相关的探针可以被用来寻找含有这种 RFLP 的克隆。这个克隆可被测序，而且它的一个末端可以被用来设计一个新的甚至非常靠近 CF 基因的探针。这些探针可以用来寻找新的克隆，并且从 D7S8“步查”到 CF 基因。经过多次重复，在标记基因周围区域的数十万 DNA 的碱基可以被测序。如果标记与感兴趣的基因连接很近，那么那个基因

也会被测序。在 CF 工程中，总长 249kb 的序列被克隆成 58 个 DNA 片段。

基因步查设计是相当复杂和单调的。一个障碍是并非所有的 DNA 区域都可以出现在克隆库中，因为基因组的某些区域在细菌中克隆时会出现不稳定。Collins 等[73]发展了染色体跳查技术，它被成功地用于定位包含 CF 基因的区域。

尽管在概念上很吸引人，染色体步查和跳查技术对于定位整个基因组实在太费力了，而定位单个基因则需要缝接处理。一个事先构建的覆盖整个基因组的图谱对于任何新基因的定位都有重要作用。

不同的指纹导致不同的作图问题。在基于和短探针杂交的指纹情况下，一个探针可能和很多克隆杂交。对于 n 个克隆和 m 个探针的图谱装配问题。杂交数据组成一个 $n\times m$ 的矩阵 $D=(d_{ij})$，其中 $d_{ij}=1$，如果克隆 C_i 包含探针 p_j，而其他情况 $d_{ij}=0$（见图 1.1）。注意数据并不表明探针在给定的克隆上出现了多少次，也不表明探针在一个克隆中出现的次序。

物理图谱最简单的近似是最短覆盖字符串问题。设 S 是一个覆盖字符探针

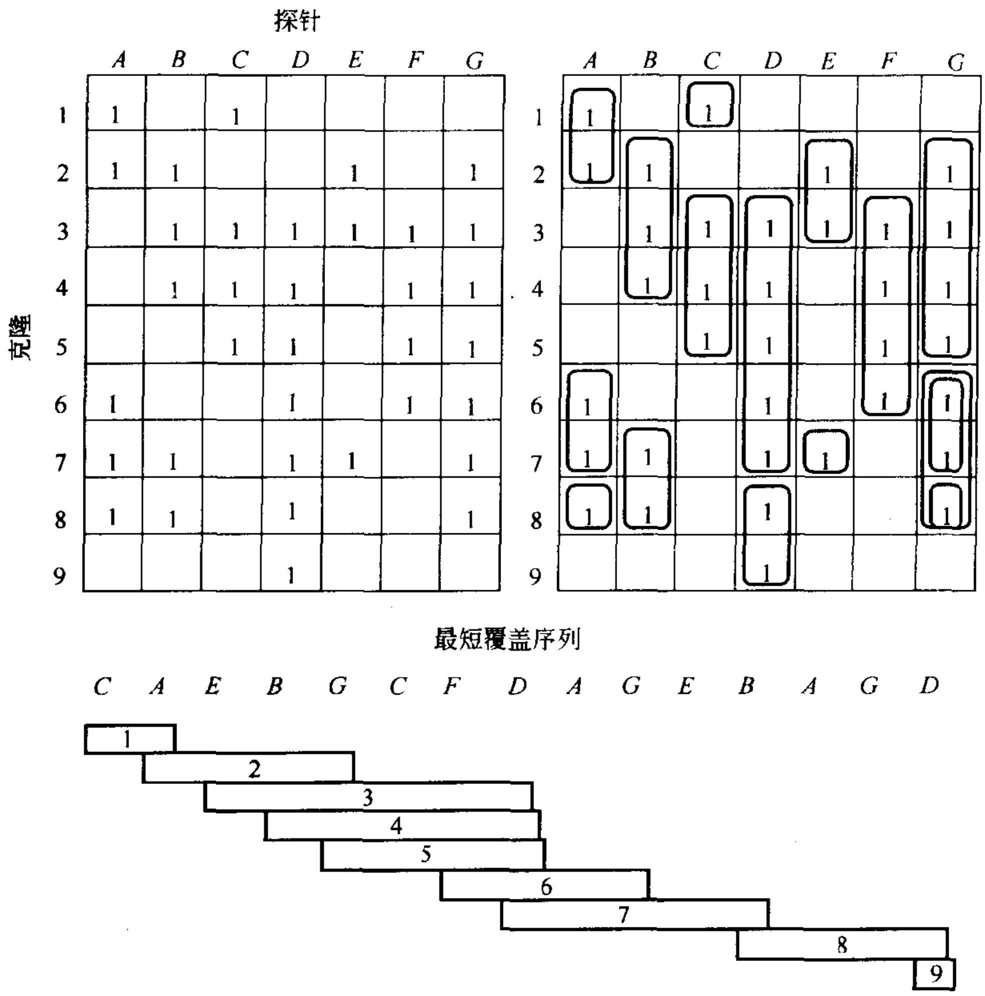


图 1.1 杂交数据和最短覆盖序列

$p_1 \cdots p_m$ 的字符序列。一个字符序列 S 包含一个克隆 C，如果存在 S 的严格地覆盖 C 的探针的集合的子序列（子序列中的探针的次序和多重性可忽略）。相应于杂交数据，图 1.1 中的序列覆盖 9 个克隆中的每一个。

最短覆盖序列问题　给定杂交数据，在探针字母表中寻找一条包含所有克隆的最短序列。

在使用探针为 DNA 作图前，生物学家构建了克隆的限制性酶切图谱并且把它们当成克隆次序的指纹。克隆的限制性酶切图谱是限制酶切片段的有序列表。如果两个克隆有共享若干连续片段的限制性酶切图谱，那么它们很可能是重叠的。运用这种策略，在 1987 年 Kohara 等[204]装配了 5×10^6 碱基对的大肠杆菌（*Escherichia coli*）基因组的限制性酶切图谱。

为了建立克隆的限制性酶切图谱，生物学家使用不同的生物化学技术来得到有关图谱的间接信息并用组合方法由这些数据重构图谱。这个问题经常可以被简述为仅当一些点间的两两距离已知时重新获得点的位置。

许多作图技术导致下述组合数学问题。如果 X 是一个在线上的点的集合。那么 ΔX 表示在 X 中的所有点间的两两距离的多重集合：$\Delta X=\{|x_1-x_2|:x_1, x_2\in X\}$。在限制性酶切图谱中给定一个子集 $E\subset\Delta X$，其相应于片段长度的实验数据，于是问题归结为只根据 E 的知识来重建 X。在部分酶切问题（PDP）中，实验数据给出所有酶切位点间的两两距离，并且 $E=\Delta X$。

部分酶切问题　给定 ΔX，重构 X。

这问题也经常被看作是计算机科学中的收费公路问题。假设你已知道所有高速公路上的每对出口之间的距离的集合。根据这些数据，你能重建高速公路的地图，即判定从高速公路的起点到每个出口的距离吗？如果你用 DNA 的限制酶剪切位点替代高速公路出口，而且如果你以这样的方法控制 DNA 的酶切，在酶切中每两个切口生成一个片段，那么 DNA 片段的大小相应于高速公路出口间的距离。

对这种看起来很琐碎的难题至今仍无多项式算法。

1.4　测序

想像一下一本书的数个拷贝被剪刀剪切成一千万块碎片。每个拷贝是用单独的方法剪切的，这样一个拷贝的碎片可能和另一个拷贝的碎片重叠。假定一百万块碎片丢失了，而其他九百万块被墨水泼过了，设法恢复原来的文本。如此做了之后你就可以感受到 DNA 测序问题像什么了。典型的测序技术允许生物学家每次实验读（300～500 个字符）的短片段（这些片段的每一个相应于一千万块碎片中的一个）。计算生物学家们只好根据这些短片段来拼装整个基因组，这个任务如同用成千上万块纸的碎片来装配书。这个问题由于不可避免的实验误差（墨

水污染）而更加复杂化。

DNA 测序最简单而又天真的近似与下述问题相同。

最短超字符串问题 给定字符串 $s_1,\cdots,s_n$ 的一个集合，寻找最短的字符串 s，使得每个 s_i 出现在 s 的一个子字符串中。

图 1.2 给出了用 0-1 字符表示的全部 8 个三字符字符串集合的两个超字符串。第一个（琐碎）超字符串通过串联这 8 个字符串而得到，第二个则是最短超字符串。这个超字符串同“聪明的小偷和译码锁”问题的解有关（小偷用来尝试所有可能的 k 个字符的密码的最少的试验次数）。

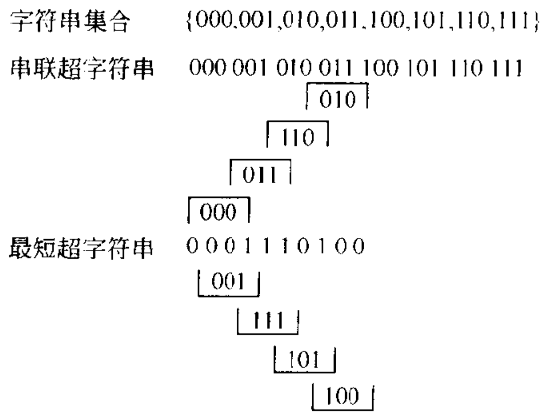


图 1.2 0-1 字符表示的全部 8 个三字符字符串集合的两个超字符串

因为最短超字符串问题已知是 NP 难题（NP-hard），所以提出了一些启发式的算法。早期的 DNA 测序算法使用了简单的贪心策略：反复地拼接具有最大重叠的字符串对直至归于一个字符串。

虽然常规的 DNA 测序现在已是非常快速和有效，但在 10 年前它却是非常耗时间并很难自动化的。1988 年有 4 个生物学家小组同时独立地提出一种被称为杂交测序（Sequencing by Hybridization，SBH）的新技术。他们提议建立一个包括数以万计的 DNA 短片段的微型 DNA 芯片（chip），也被称为 DNA 阵列（array），它们可像芯片的存储器那样工作。这些短的片段每一个揭示了有关一个未知 DNA 片段的某些信息，把所有这些片段的信息组合在一起就可解决 DNA 测序难题。但是在 1988 年几乎没有人相信这个主意能够成功：无论生物化学问题（在阵列的表面合成数以万计的 DNA 短片段）还是组合问题（通过阵列的输出重构序列）看起来都太复杂了。现在，构建具有成千上万个探针的 DNA 阵列已经成为一种工业。

给出一条未知核苷酸序列的 DNA 片段，一个 DNA 阵列提供 l 元组（l-tuple）的组成，即关于包含在此片段中的所有长度为 l 的子字符串的信息（这些子字符串的位置是未知的）。

杂交测序问题 根据字符串的 l 元组组成重构一个字符串。

虽然 DNA 阵列最早是为了 DNA 测序而发明的，但用这种技术仅测定了很少的片段[90]。出现的问题是杂交过程中的失真导致了 l 元组组成的错误。如同生物学中常常发生的，DNA 阵列先被证明它对发明出来要解决的原始问题并不成功，但在功能基因组和突变检测方面却获得了不同的应用。

虽然常规的 DNA 测序和 SBH 是非常不同的方法，但相应的计算问题却是

相同的。事实上，SBH 是最短超字符串问题的一种特殊情形，当所有的字符串 $s_1,\cdots,s_n$ 表示固定长度 s 的全部子字符串的集合。然而，同最短超字符串问题相反，SBH 问题存在一种简单的线性时间的算法。

1.5 相似性搜寻

在测序之后，生物学家们对找到的基因的功能没有任何主意。为了找到基因功能的线索，他们试图寻找新测序的基因和已知功能的先前测序的基因的相似之处。通过相似性搜寻而产生的生物学发现的一个惊人例子发生在 1984 年，当时科学家们使用简单的计算技术比较最新发现的致癌基因 *v-sys* 和所有已知的基因。使他们惊奇的是，这个致癌基因与一个同生长和发育相关的基因匹配。突然这变得很清楚了，癌可能是由于普通的生长基因在不适当的时间出现变化而引起的[89,353]。

1879 年 Lewis Carroll 给名利场（Vanity Fair）的读者提出下述难题：通过一系列英语单词的过渡，将一个英语单词转变为另一个单词，其中每个单词只能和下一个有一个字母的不同。将 head 变为 tail 只需要四个过渡单词：head→heal→teal→tell→tall→tail。1966 年，Levenshtein[219] 引入了字符串之间编辑距离的概念，就是将一个字符串转变为另一个字符串所需要的最少的基本操作的数目，其中基本的操作是插入一个字符，删除一个字符和将一个字符用另一个替换。大多数的序列比较算法与计算由上述或稍有不同的基本操作步骤所产生的编辑距离有关。

因为 DNA 的突变表征自然的演化过程，所以编辑距离是 DNA 片段间相似性的一个自然度量。DNA 序列之间的相似性可以作为有共同进化起源的线索（就像人类和黑猩猩的球蛋白基因的相似性）或者有相同功能的线索（像 *v-sys* 致癌基因和生长刺激激素的相似之处）。

如果编辑操作局限于插入和删除（没有替换），那么编辑距离问题就相当于最长共同子序列（LCS）问题。给出两个字符串 $V=v_1\cdots v_n$ 和 $W=w_1\cdots w_m$，一个长度为 k 的 V 和 W 的共同子序列是一个指标序列 $1\leqslant i_1<\cdots<i_k\leqslant n$ 和 $1\leqslant j_1<\cdots<j_k\leqslant m$，使得

$$v_{i_t}=\mathrm{w}_{j_t} \quad 当\ 1\leqslant t\leqslant k$$

设 LCS(V,W) 为 V 和 W 的最长共同子序列（LCS）的长度。例如，LCS(ATCTGAT,TGCATA)＝4（形成 LCS 的字母用黑体）。显然，$n+m-2\mathrm{LCS}(V,W)$是将 V 变为 W 的最少的插入和删除的步骤。

最长共同子序列问题 给出两个字符串，寻找最长的共同子序列。

当囊肿性纤维化基因的周围的区域已经被测序，生物学家将它和所有已知的基因数据比较并且发现长度约为 6500 个核苷酸的片段与一类称为 ATP（三磷酸

腺苷）结合蛋白的序列间的某些相似。这类蛋白早已被发现且已知是多次跨越细胞膜，其功能是作为离子跨膜传输的通道。这看起来像是 CF 基因可能的功能，给出这个事实是这种疾病产生异常的分泌。这个相似性也指出两个保守的 ATP 结合位点（ATP 蛋白为细胞中的许多反应提供能量）并使有缺陷的 CF 基因造成的机制上的损害清楚明白地显示了出来。其结果是囊肿性纤维化基因被称为囊肿性纤维化跨膜传导调节蛋白。

1.6 基因预测

知道近似的基因位置仍不能找到基因本身。例如，亨廷顿疾病基因在 1983 就已经被定位，但是直到 1993 年才找到。与之成对照的是 CF 基因 1985 年被定位，而在 1989 年找到。

在最简单的生命形式中，例如细菌，用 DNA 记录的基因是连续的字符串。而对于人类（或其他哺乳动物）情形并非如此简单。一个人类基因大概由 2000 个字符组成，通常被断裂成若干称为外显子的子片段。这些外显子似乎笨拙地、看起来是任意地散布在长度为百万字符的染色体 DNA 的一个区域之中。一个典型的人类基因可能有 10 个外显子甚至更多。同乳腺癌有关的 *BRCA*1 基因有 27 个外显子。

这种情形可与杂志的文章做比较，开始是第 1 页，接着是第 13 页，然后再在 43 页，51 页，53 页，74 页，80 页和 91 页中间则刊载广告和登载其他文章。人类基因组的 97％是广告或所谓的“垃圾”DNA，不知道为什么会出现这些跳跃及其目的。

跳跃在不同的物种之间是不一致的，一篇“文章”在遗传杂志的昆虫版中将与同一篇文章的蠕虫版有不同的印刷。它们的页码标记方法是完全不同的，显示于人类版本中单页的信息可能在小麦版本中被分成两页，或者反之亦然。有关的基因本身是不同的。小鼠版本的基因用小鼠语言记录，人类的基因是用人类的语言记录的。这有点像德语和英语，很多单词是相似的，但是也有许多是不同的。

在新测序的 DNA 序列中预测新的基因是一个困难的问题。决定哪些是广告哪些是故事的方法大都依靠统计。继续杂志模拟，这就像是仔细检查过去的杂志并找到人类基因“故事”不太可能包含的某些短语，例如“出售”、电话号码和钱币符号。与此相反，基因预测的组合方法则把先前已测定的基因作为识别新测序基因的模板。替代外显子的统计性质，这种方法试图解决一个组合难题：在基因组序列中寻找一组序列块（候选外显子），它们的串联（拼接）与一个已知的蛋白质相吻合。图 1.3 对一个“基因组”序列难题做了说明。

'twas brilliant thrilling morning and the slimy hellish lithe doves
gyrated and gambled nimbly in the waves

不同的序列块拼接成 Lewis Carroll 的有名的“目标蛋白质”：

't was brillig, and the slithy toves did gyre and gimble in the wabe

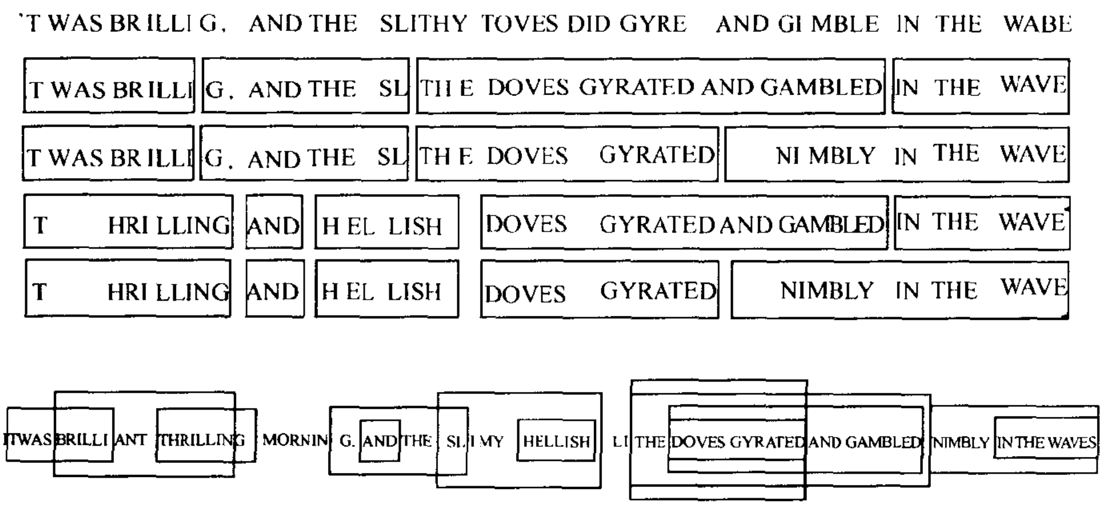


图 1.3 剪接联配问题：与 Lewis Carroll 的“目标蛋白质”最佳匹配的序列块装配

这个组合难题导致下述问题。

剪接联配问题 设 G 为一个基因组序列的字符串，T 是目标序列的字符串，$\mathscr{B}$ 是 G 的一组子字符串。给定 G，T 和 $\mathscr{B}$，从 $\mathscr{B}$ 中找出一组非重叠的字符串，它们的串联序列同目标序列最佳匹配（即 $\mathscr{B}$ 的字符串块的所有组合中，这些字符串的串联序列与目标序列的编辑距离是最小的）。

1.7 突变分析

当一个感兴趣的基因已经被测序，但对那个基因的结构却一无所知，于是，探索基因的结构成了基因搜寻中的挑战之一。在囊肿性纤维化的例子中，基因预测和序列相似性对该基因提供了一些线索，但是没有排除其他的候选基因。特别是，其他三个片段都是可能的。如果一个可疑的基因是真正的疾病基因，那么受疾病侵袭的个体在这个基因上会发生突变。取自不同个体中的这一基因将被重新测序以验证这个假设。CF 基因中的一个突变（三个核苷酸的缺失，引起一个氨基酸的缺失）被发现在受侵袭的个体中是共同的。这是一个先例，PCR（聚合酶链反应）引物被用于从大量的个体中筛选这种突变。这种突变在 70%的囊肿性纤维化病人中被发现，因而令人信服地证实了是它导致囊肿性纤维化。各种各样不同的突变组成了额外 30%有缺陷的囊肿性纤维化基因，这使得囊肿性纤维化的医学诊断更加困难。为囊肿性纤维化专门设计的 DNA 阵列可能对突变的群体筛选很有效。

相似性搜寻、基因识别和突变分析提出了一系列的统计问题。如果两个序列有 45%相似，它们是否可能是真正相关的，或者只是偶然？基因中经常发现，

在 DNA 片段中常出现高频率 CG 二核苷酸（CG 岛）。特别是囊肿性纤维化基因正位于 CG 岛内。怎么样水平的 CG 含量表示一个 CG 岛，而哪些又只是偶然？下面给出相应的统计例子。

LCS 的预期长度问题 寻找两个长度为 n 的任意字符串的 LCS 的预期长度。

字符串统计问题 给定一个字符串，找出其在一个任意文本中出现次数的期望和方差。

1.8 比较基因组

就像在囊肿性纤维化基因寻找中已经看到的，人类基因的搜寻是个缓慢而费力的过程。但是，动物中相同的遗传病症的遗传学研究常可加快这个进程。

瓦登伯格（Waardenburg）综合征是导致听力丧失和色素发育不全的先天性遗传病症。遗传图谱将 Waardenburg 综合征基因的搜寻限定在人类第二条染色体上，但是它的准确位置仍然不知道。存在直接关注第二条染色体的另一线索。很长一段时间，饲养者仔细观察了突变体小鼠，指明了斑点病，患病的小鼠带有白色斑点，这种病被认为同 Waardenburg 综合征相似。通过饲养（饲养小鼠比人简单），斑点病的基因被定位在小鼠的第二条染色体上。在基因作图过程中，终于清楚了在两个物种中存在相互间紧密联系的基因群。基因组的移动在进化过程中并没有结束；在多个染色体重新排列发生时遗传物质的组件仍保持完整。例如，构成人类的第二条染色体的一些片段同小鼠位于 1 条，2 条，6 条，8 条，11 条，12 条和 17 条染色体上的 DNA 片段相似（见图 1.4）。因此，小鼠基因的定位常可提供相关的人类基因的位置线索。

尽管在外表和习惯上人和小鼠有所不同，但是它们总的来说还是非常相似的。在先驱的文章上，Nadeau 和 Taylor 在 1984 年[248]估计自从 8 亿年前人类和小鼠分叉之后极少地发生了基因组重排（178±39）。小鼠和人的基因组可以看成是大概 200 个片段的集合，与人类的相比小鼠的片段有所移动（重排）。如果一个小鼠的基因定位于上述片段之一，那么相应的人类基因会定位在与这个小鼠基因连接的一个染色体片段上。一张比较小鼠和人的遗传图谱可根据给定的相关的小鼠基因的定位而给出一个人类基因的位置。

基因组重排是相当普遍的染色体异常，这同一些遗传疾病例如唐氏（Down）综合征有关。通常，基因组重排是无症状的：估计有 0.2%的个体携带无症状的染色体重排。

基因组重排的分子生物学分析是由 Dobzhansky 和 Sturtevant 在 1938 年[87]开创的，他们发表了里程碑式的论文，文中对果蝇基因组提出了含 17 个倒位的重排方案。就最简单的形式而言，将一种基因组转换为另一种基因组的基因组重排可用寻找最小转换次数的组合问题来模拟。生物体中的基因顺序可用排列

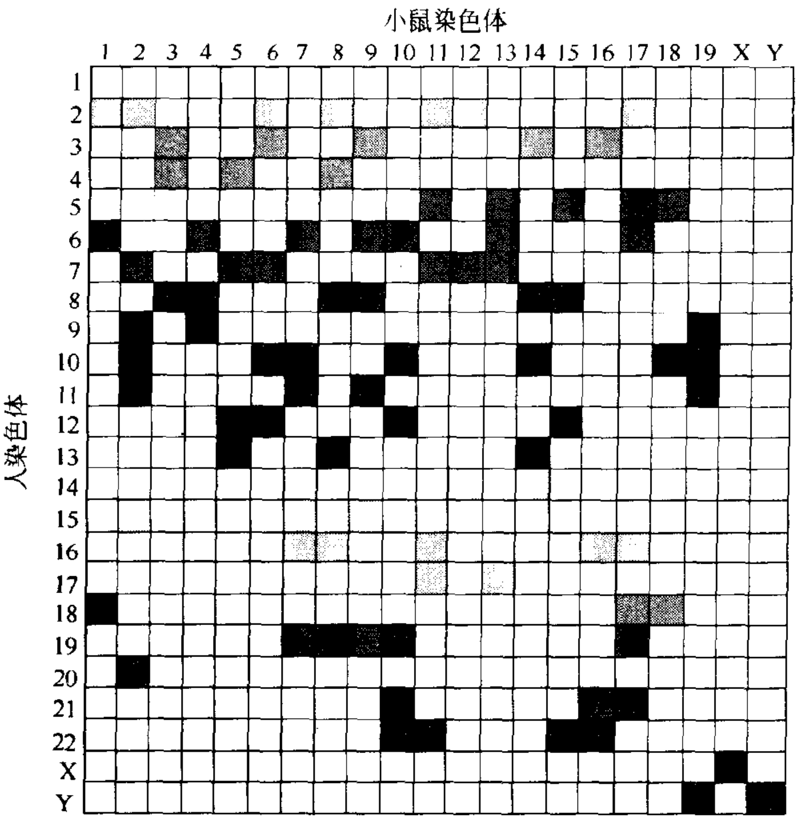


图 1.4 人和小鼠比较的物理图谱

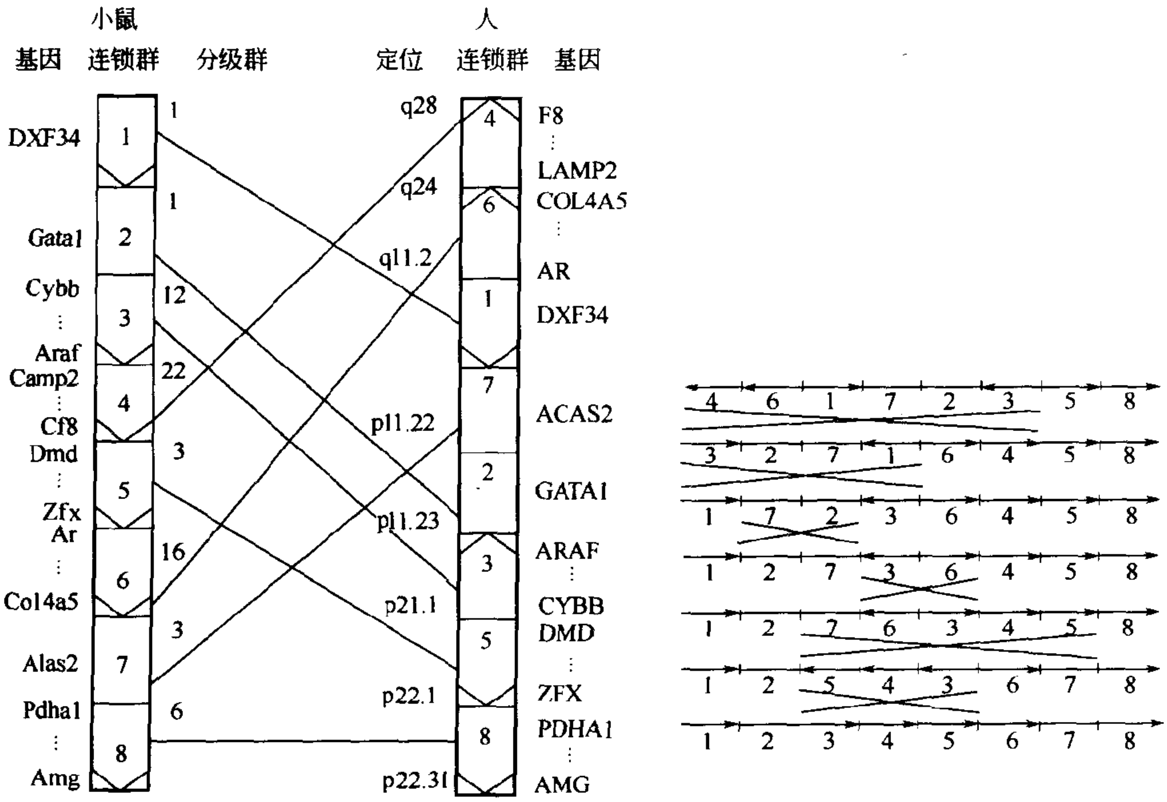


图 1.5 人的 X 染色体“转换”为小鼠的 X 染色体

$\pi=\pi_1\pi_2\cdots\pi_n$ 来表示。反转 $\rho(i,j)$ 具有反转基因 $\pi_i\pi_{i+1}\cdots\pi_j$ 次序的作用，它将 $\pi=\pi_1\cdots\pi_{i-1}\pi_i\cdots\pi_j\pi_{j+1}\cdots\pi_n$ 转变为 $\pi\cdot\rho(i,j)=\pi_1\cdots\pi_{i-1}\pi_j\cdots\pi_i\pi_{j+1}\cdots\pi_n$。图 1.5 展示了一个将人的 X 染色体转换为小鼠的 X 染色体的重排方案。

反转距离问题 给定序列 π 和 σ，寻找一系列的反转 $\rho_1,\rho_2,\cdots,\rho_t$ 使得 $\pi\cdot\rho_1\cdot\rho_2\cdots\rho_t=\sigma$，且要求 t 为最小。

1.9 蛋白质组学

在许多生物体的发育过程中，细胞在特定的时期死亡作为正常生命进程的一部分，被称为细胞编程死亡。死亡可能因为无法获得生存因素而发生，也可能因特定基因的表达而启动。例如，在一个线虫的生长过程中，神经系统中个别细胞的死亡可由一些基因的突变来预防。这些基因的功能正在积极地研究中。然而，前面介绍的基于 DNA 的方法已不适用于寻找涉及细胞编程死亡的基因。

细胞死亡组织是一个由许多基因组成的复杂系统。虽然同这些候选基因相关的许多蛋白质已被鉴定，但它们在细胞编程死亡中的角色和相互作用的方式还了解得很少。困难在于这些候选基因的 DNA 很难分离，至少比相应的蛋白质难。然而，对于蛋白质测序目前仍然没有可靠的方法，因此这些候选基因的序列到目前仍然未知。

最近，用于蛋白质测序的新的质谱测定方法已经出现，它可对许多涉及细胞编程死亡的蛋白质进行测序。在 1996 年，蛋白质测序导致了 FLICE 蛋白的鉴定，它是包含死亡诱导信号的复合蛋白质[244]。在这种情形，基因搜寻与其从 DNA 测序开始，不如从蛋白质测序开始，而这导致随后克隆 FLICE 基因。这种异常灵敏的质谱测定法的出现开辟了一条蛋白质测序的新的实验和计算的道路，也使这项技术成为许多领域所采用的方法。

蛋白质测序有很长的极度迷人的质谱线[182]。然而，仅在当前，随着自动质谱测定系统和从头（*de novo*）算法的发展，高通量的蛋白质测序才成为现实，并终于打开了“蛋白质组测序”之门。目前，大多数蛋白质是通过数据库的搜寻来鉴定的[97,230]，这是依靠“在书后寻找答案”的能力。虽然数据库搜索在大量已测序的基因组中是非常有用的，但试图发现一个新基因的生物学家需要从头算法更甚于数据库搜索算法。

在几秒之内，质谱仪可将肽分割成许多断片（离子）并测量它们的质量。获得的质量组形成一个肽的谱。肽测序问题是根据给出的谱重构这个肽。对于一个“理想的”断裂处理方法和一台“理想的”质谱仪，肽测序问题是简单的。在实践中，因为谱图难以解释，所以从头肽测序仍是一个尚未解决的问题。

在最简单的形态，用质谱测定方法作蛋白质测序相应于下述问题。设 A 是氨基酸的集合，分子质量为 $m(a)$，$a\in A$。一个（亲本）肽 $P=p_1,\cdots,p_n$ 是氨

基酸序列，且肽 P 的质量为 $m(P)=\sum m(p_i)$。部分肽 $P'\subset P$ 是 P 的一个质量为 $\sum_{i\leqslant t\leqslant j} m(p_t)$ 的子序列 $p_i\cdots p_j$。肽 P 的理论谱 $E(P)$ 是其部分肽的质量的集合。一个（实验）谱 $S=\{s_1, \cdots, s_m\}$ 是离子（片段）质量的集合。谱 S 和肽 P 之间的匹配是实验和理论谱共有的质量数。

肽测序问题 给出谱 S 和亲本质量 m，寻找一个与谱 S 最大匹配的质量为 m 的肽。

2 限制酶切作图

2.1 引言

1970 年，Hamilton Smith 发现限制酶 HindⅡ在出现序列 GTGCAC 或 GT-TAAC 的每一处切割 DNA 分子（Smith and Wilcox，1970[319]）。不久以后，Danna 等 1973 年[80]构造了猿猴病毒 40 DNA 的第一个限制酶切图谱。从那以后，表示具有限制酶切位点的 DNA 分子的限制酶切图谱（有时也称物理图谱）成了分子生物学中的基本数据组织。

为了建立一个限制酶切图谱，生物学家采用不同的生物化学技术导出有关图谱的间接信息，且用组合方法根据这些数据重建图谱。已经有了几个限制酶切作图的实验方法，各有各的优点，也各有各的缺陷。它们导致不同的组合问题，当只知道点间两两距离时，这些问题常被归结为找回点的位置。

大部分限制酶切作图问题可以表述为下述问题。如果 X 是线上的点集，ΔX 表示为 X 中所有点间两两距离的多重集合：$\Delta X=\{|x_1-x_2|: x_1, x_2\in X\}$。在限制酶切作图中，某一子集 $E\subset\Delta X$ 是已知的相应于片段长度的实验数据，问题是要根据 E 重构 X。

对于部分酶切问题（PDP），实验提供了限制酶切位点间的所有成对距离（$E=\Delta X$）的数据。在这种方法中，DNA 以每两个切割位点形成片段的方式进行酶切。还没有 PDP 的多项式算法。困难在于由 ΔX 重构的 X 可能不是惟一的：两个多重集合 X 和 Y 是同效的（homometric），如果 $\Delta X=\Delta Y$。例如，X，$-X$（X 的反射集合），以及对每一个数 a 的 $X+a$（X 的平移）是同效的。有一些平凡的非惟一性例子；例如，集合 {0，1，3，8，9，11，13，15} 和 {0，1，3，4，5，7，12，13，15} 是同效的，而且不可以通过反射和平移进行相互转换(强同效集)。在 1982 年，Rosenblatt 和 Seymour[289]研究了强同效集，且给出了基于多项式因式分解的求解 PDP 的伪多项式算法。后来，Skiena 等[314]提出了一个简单的向后追踪算法（backtracking algorithm），实际应用效果很好，但有时可能需要指数时间。

对所有实际长度的输入，向后追踪算法容易解决 PDP 问题。但是，在生物实验室里，PDP 从来就不是一个令人满意的作图方法，因为当每两个位点中形成一个切割位点时这种方法很难酶切 DNA。

双酶切是一个比部分酶切简单得多的实验作图技术。在这个方法中，生

物学家通过DNA完全酶切来定位两个限制酶的酶切位点的位置，在这方法中仅生成相邻切割位点间的片段。建立这样一个图谱的方法是，先单独用两个酶中的每一个进行DNA完全酶切，再用两个酶一起酶切，然后测量酶切生成的片段长度（不是顺序）。这种由片段长度数据确定切割位置的问题称为双酶切问题（DDP）。

对有 n 个元素的任意集合 X，记 δX 为 X 中相邻元素间的 $n-1$ 个距离的集合。在双酶切问题中，一个多重集 $X\subset[0, t]$ 被分拆为两个子集 $X=A\cup B$，其中 $0\in A, B$，$t\in A, B$，实验提供了3个长度集合：δA，δB 和 δX（A 和 B 表示单酶切，X 表示双酶切）。双酶切问题就是要根据这些数据重构 A 和 B。

解决双酶切问题的第一次尝试[329]绝不能说是成功的。其原因是随着位点个数的增长，潜在的图谱数量和DDP的计算复杂性也快速增长。问题由于实验错误而复杂化，所有DDP算法都遇到了计算困难，甚至是对于每个限制酶仅有小于10个酶切位点的小图谱。

Goldstein和Waterman在1987年[130]证明了DDP是NP完全的，表明DDP的解是随着酶切位点的增长而呈指数增长的。当然，NP完全性和解的指数增长是DDP算法的瓶颈。然而，Schmitt和Waterman在1991年[309]注意到，虽然随着酶切位点的增长解的数目增长很快，但是大部分解都非常相似（通过简单的转换方式可以互相转化）。既然作图算法产生了许多"非常类似的图谱"，就有理由将整个物理图谱集分拆为等价类，且在每个等价类中仅产生一个基图谱。接着，所有的解可以通过简单的转换方式从基图谱中产生。如果等价类的数目显著地少于物理图谱的数目，那么这种方法将会减少DDP算法的计算时间。

Schmitt和Waterman[309]在这个方向上跨出了第一步，引进了关于物理图谱的等价关系。同一个等价类的所有图谱都可用盒转换方式（cassette transformation）进行互相转化。然而，DDP中所有等价类产生的构造问题仍未解决，等价图谱的转换算法也还未知。Pevzner在1995年[267]给出了一个物理图谱等价转换的特征化定理，描述了如何产生DDP问题的所有解。这个结果是建立在DDP解和边着色图中的交错欧拉圈之间的关系基础上。

正像所看到的，PDP的组合算法在实际应用中非常快，但是PDP实验数据很难获得。相反，DDP的实验很简单，但组合算法太慢。这就是为什么限制性酶切作图至今还不是一个很普遍的实验技术的原因。

2.2 双酶切问题

如图2.1所示"DNA"用限制性酶 A 和 B 切割。Danna等[80]在建立第一个

物理图谱的时候，还没有能直接找到切割位置的实验技术。然而，已经可以用称之为凝胶电泳的实验技术来测定限制片段的大小（但不是顺序!）。虽然凝胶电泳用两个限制酶 A 和 B 做实验（见图 2.1），生物学家得到了 A 的限制片段 2，3，4 和 B 的限制片段 1，3，5 的大小信息，但是有许多排序（图谱）与这些大小相对应（图 2.2 中给出了其中的两个）。为了找出图 2.2 中哪一个图谱是正确的，生物学家用双酶切 $A+B$——用两个酶 A 和 B 切割 DNA。图 2.2 中的两个图谱产生了同样的单酶切 A 和 B，但不同的双酶切 $A+B$（1，1，2，2，3 和 1，1，1，2，4）。符合实验数据的双酶切所对应的是正确的图谱。为了找到正确的物理图谱，双酶切问题给出三个片段“堆”：A，B 和 $A+B$（见图 2.3）。

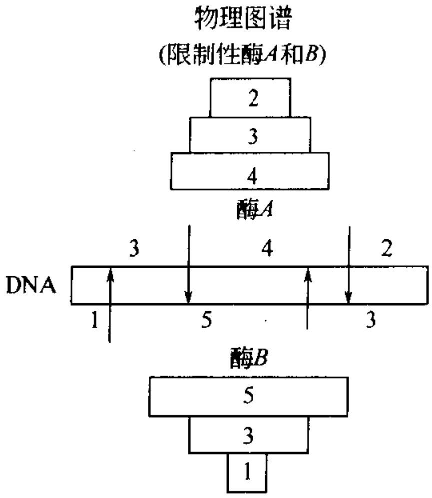


图 2.1 两个限制酶的物理图谱
凝胶电泳提供了限制片段大小（但不是顺序）信息

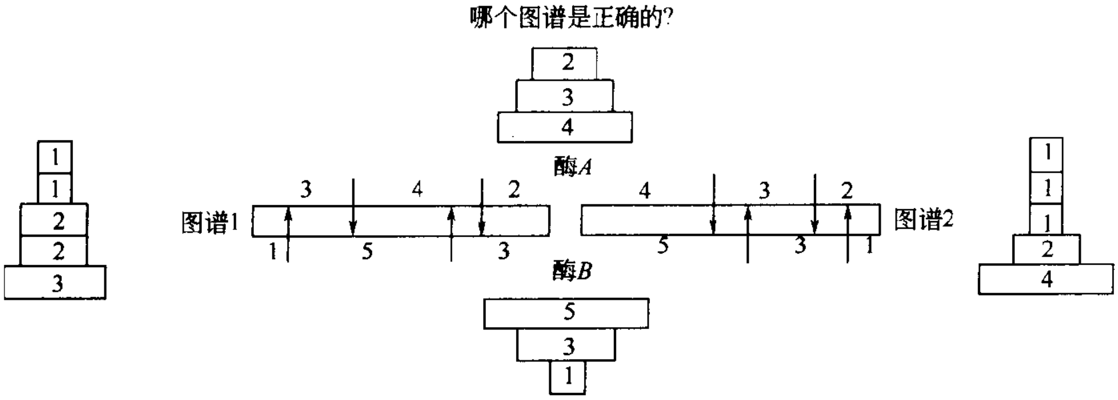
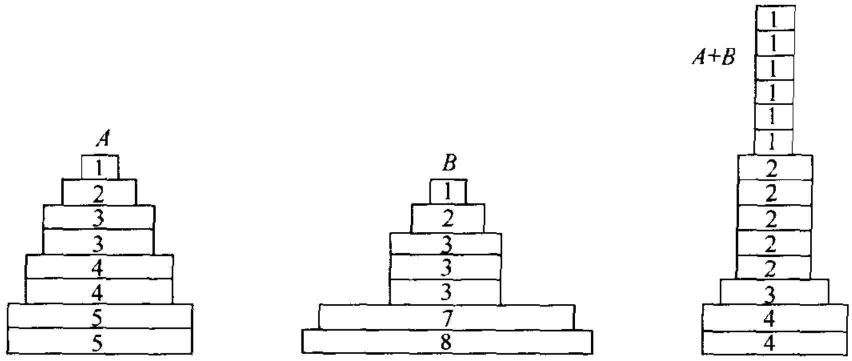


图 2.2 A 和 B 的数据生物学家并不能找到真实的图谱
$A+B$ 数据有助于找到正确的图谱

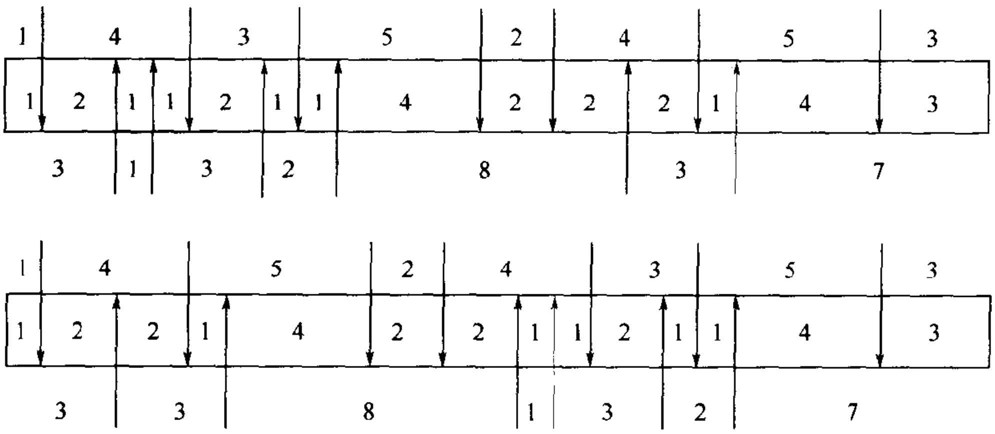
2.3 双酶切问题的多重解

图 2.3 展示了双酶切问题的两个解。虽然它们看上去很不相同，但它们可以用一个称为盒式交换（cassette exchange）（见图 2.4）的简单算子进行互相转换。多重解的另一个例子在图 2.5 中给出。虽然这些解不可以用盒式交换进行相互转换，但它们可以用另一种称为盒式反射的运算进行互相转换。一个令人感到惊奇的结果是，在某种意义下，这两种简单的操作为双酶切问题中的任意两个“相似”解提供了充分的转换方式。

一个物理图谱可表示成由单酶切 $A_1, \cdots, A_n$ 和 $B_1, \cdots, B_m$ 以及双酶切



(a) 双酶切问题
给定A, B和$A+B$, 寻找物理图谱



(b) 多重DDP解

图 2.3 双酶切问题可有多重解

C_1，…，C_l（见图 2.4）片段的顺序序列。在区间 $I=[i, j]$ 中，$1\leqslant i\leqslant j\leqslant l$，定义 $I_C=\{C_k: i\leqslant k\leqslant j\}$ 为 C_i 和 C_j 间的片段集合。由 I_C 定义的盒就是片段（I_A，I_B）的成对集，其中 I_A 和 I_B 分别是包含了 I_C 中一个片段的 A 和 B 的所有片段集合。令 m_A 和 m_B 分别是 I_A 和 I_B 中最左边的片段的起始位置，（I_A，I_B）的左边重叠指距离 m_A-m_B。可同样定义（I_A，I_B）的右边重叠，用“结束”和“最右边”替代上面定义中的“起始”和“最左边”。

假设 DDP 解中的两个盒有相同的左重叠和右重叠。如果这两个盒不相交（没有公共片段），那么它们可以如图 2.4 所示那样交换，并得到一个新的 DDP 解。同样，如果一个盒（I_A，I_B）的左右重叠有相同大小，但符号不同，那么盒可以如图 2.6 所示的那样进行反射，并得到 DDP 的一个新解。

Schmitt 和 Waterman[309] 提出了如何用盒式变换方式进行图谱相互转换的问题。下面一节中介绍图论技术以分析盒式变换的组合，并回答 Schmitt-Waterman 问题。

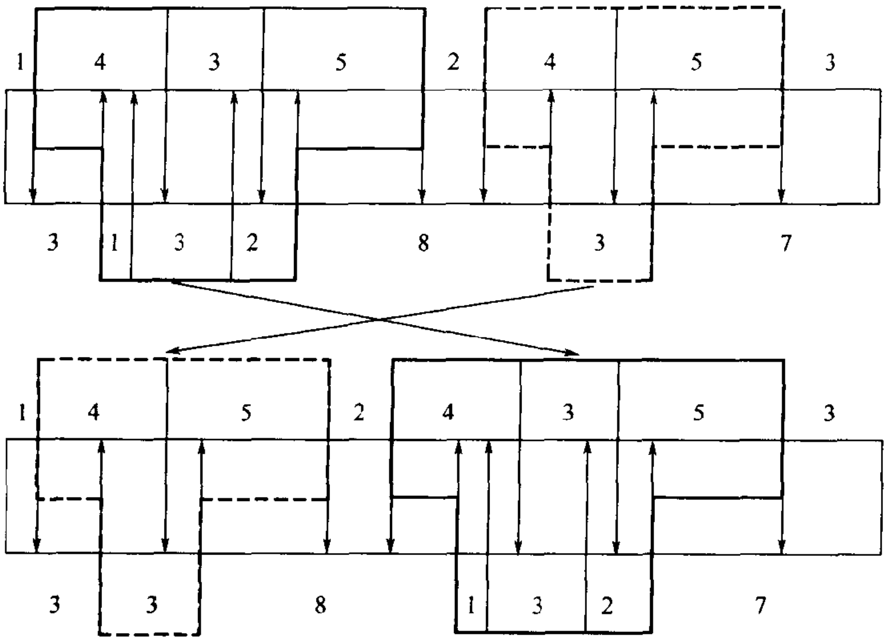
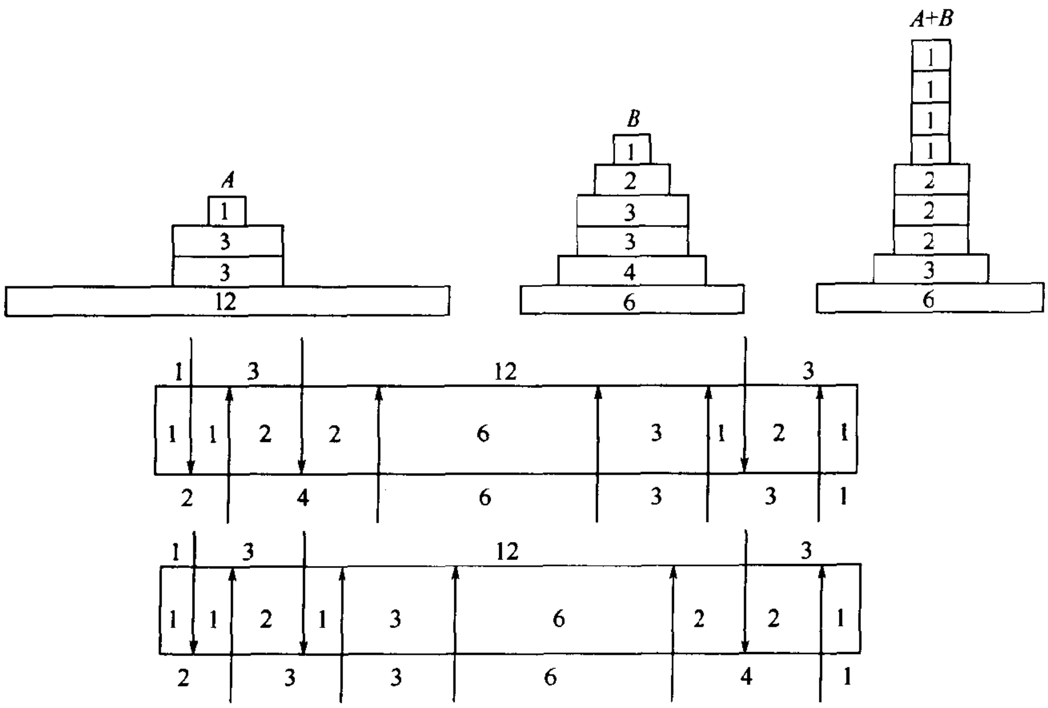


图 2.4 盒式交换

限制酶$A(\{1, 4, 3, 5, 2, 4, 5, 3\})$、限制酶$B(\{3, 1, 3, 2, 8, 3, 7\})$和限制酶$A+B=C(\{1, 2, 1, 1, 2, 1, 1, 4, 2, 2, 2, 1, 4, 3\})$的酶切片段大小的顺序序列。区间$I=[3, 7]$指长度为1，1，2，1，1的双酶切片段$I_C=\{C_3, C_4, C_5, C_6, C_7\}$的集合。$I_C$定义一个盒$(I_A, I_B)$，其中$I_A=\{A_2, A_3, A_4\}=\{4, 3, 5\}$，而$I_B=\{B_2, B_3, B_4\}=\{1, 3, 2\}$。$(I_A, I_B)$的左边重叠于$m_A-m_B=1-3=-2$。$(I_A, I_B)$的右边重叠等于$13-9=4$



多重DDP解(另一个例子)

图 2.5 不能由盒式交换而相互转换的多重 DDP 解

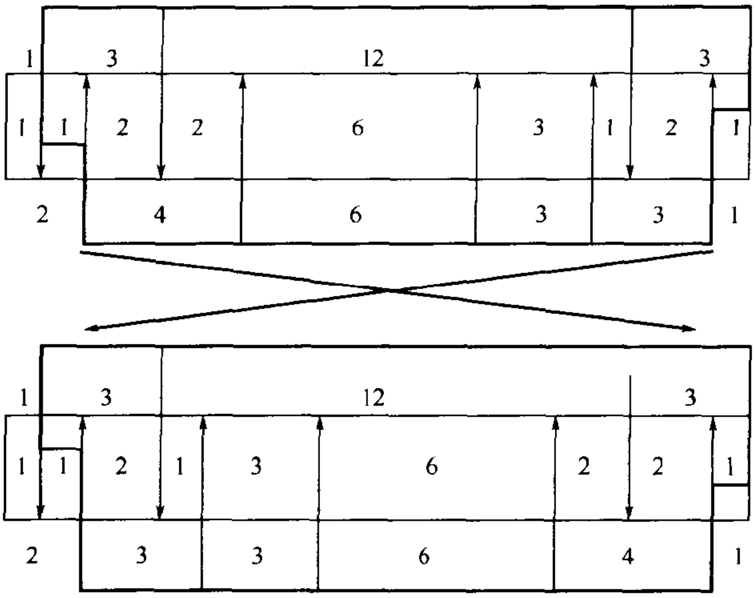


图 2.6 盒式反射

左和右重叠大小相同，符号不同

2.4 着色图中的交错圈

考虑一个无向图 $G(V, E)$，它的边集合 E 中的边用 l 种颜色着色。顶点序列 $P=x_1x_2\cdots x_m$ 称为 G 中的一条路径，如果 $(x_i, x_{i+1})\in E$，$1\leqslant i\leqslant m-1$。路径 P 称为圈，如果 $x_1=x_m$。路径和圈可以是顶点自相交（vertex self-intersecting）的。记作 $P^-=x_mx_{m-1}\cdots x_1$。

G 中的一条路径（圈）称为交错的，如果这个路径（圈）中每两个连续的边 (x_i, x_{i+1}) 和 (x_{i+1}, x_{i+2}) 的颜色不同。如果 P 是一个圈，则认为 (x_{m-1}, x_m) 和 (x_1, x_2) 是连续的边。G 中的一条路径（圈）P 称为欧拉圈（Eulerian cycle)，如果对每个 $e\in E$，P 只经过一次。令 $d_c(v)$ 是与 v 相关的 E 中用 c 着色的边的数目，且 $d(v)=\sum_{c=1}^{l}d_c(v)$ 是图 G 中顶点 v 的度。图 G 中顶点 v 称为是平衡的，如果 $\max_c d_c(v)\leqslant d(v)/2$。一个平衡图的每一个顶点都是平衡的。

定理 2.1[206] 令 G 是一个着色连通图，它的顶点具有偶次度。当且仅当 G 是平衡的，则 G 中有一个交错欧拉圈。

证明 考虑 G 中的一个交错欧拉圈，将与顶点 v 相关的 $d(v)$ 条边分拆成 $d(v)/2$ 对，使得同一对中的两条边有不同颜色（对每一个平衡顶点都可满足）。从 G 中的任一条边开始，生成一条轨迹 C_1，每一步选择一条边与轨迹中的最后一条边配对。当所选的与轨迹中最后一条边配对的边已经被选过时，这个过程就结束。由于 G 中的每一个顶点均有偶次度，每条这样的轨迹从 v 开始，又到 v 结束。幸运的话，这条轨迹将是欧拉圈，但若不是，则它肯定包含一个结点 w，

且仍有一些未遍历的边。由于未遍历边的图是平衡的，可以从 w 开始，采用同样的步骤，经未遍历的边生成另一条轨迹 C_2。现在，可以把轨迹 C_1 和轨迹 C_2 进行如下组合：把轨迹 C_2 插入到 w 到达点处的轨迹 C_1 中。在这个过程中必须注意保持顶点 w 处颜色的交错。可以看到，如果正向插入轨迹 C_2，会破坏颜色的交错性，而反向的插入则保持了颜色的交错性。重复这个过程可以最终得到一个交错欧拉圈。 ■

下面的 Kotzig 定理的推论将被用到。

引理 2.1 令 G 是一个双色连通图。当且仅当对 G 中每一个顶点有 $d_1(v)=d_2(v)$，则 G 中存在一个交错欧拉圈。

2.5 交错欧拉圈的变换

这一节中，将介绍交错路径的顺序变换，并证明双色图中每两个交错欧拉圈可以通过顺序变换相互转化。这个结果隐含了 Schmitt-Waterman 盒式变换的特征。

令 $F=\cdots x\cdots y\cdots x\cdots y\cdots$ 是双色图 G 中的一条交错路径。顶点 x 和 y 把 F 分拆为五条子路径 $F=F_1F_2F_3F_4F_5$（见图 2.7）。如果 F^* 是一条交错路径，变换 $F=F_1F_2F_3F_4F_5 \longrightarrow F^*=F_1F_4F_3F_2F_5$ 称为一个顺序交换。

令 $F=\cdots x\cdots x\cdots$ 是双色图 G 中的一条交错路径。顶点 x 把 F 分拆为三条子路径 $F=F_1F_2F_3$（见图 2.8）。如果 F^* 是一条交错路径，变换 $F=F_1F_2F_3 \longrightarrow F^*=F_1\overline{F_2}F_3$ 称为一个顺序反射。显然，当且仅当 F_2 是一个奇数圈时，在双色图 G 中存在顺序反射 $F\rightarrow F^*$。

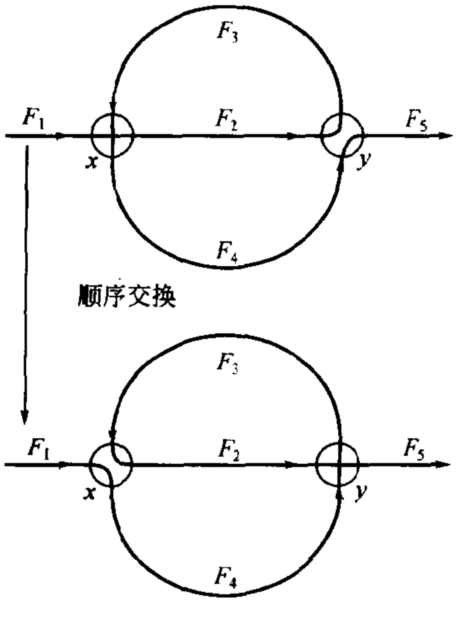


图 2.7 顺序交换

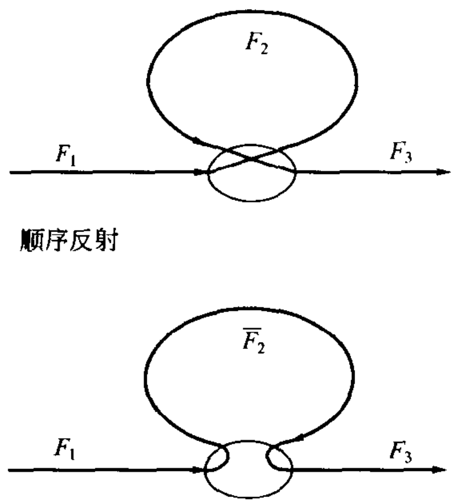
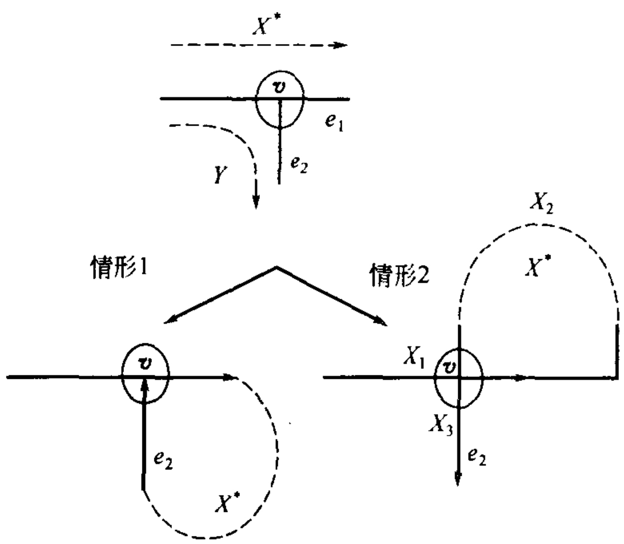


图 2.8 顺序反射

定理 2.2　双色图中的任意两个交错欧拉圈可以通过一系列顺序变换（交换和反射）进行互相转化。

证明　令 X 和 Y 是 G 中两个交错欧拉圈。考虑在 X 中用所有可能的系列顺序变换得到交错欧拉圈集合 $\mathcal{C}$。令 $X^*=x_1\cdots x_m$ 是 $\mathcal{C}$ 中具有 $Y=y_1\cdots y_m$ 最长公共前缀的圈，即对 $l\leqslant m$ 有 $x_1\cdots x_l=y_1\cdots y_l$。如果 $l=m$，定理成立；否则，令 $v=x_l=y_l$［即 $e_1=(v,x_{l+1})$，$e_2=(v,y_{l+1})$ 分别是 X^* 和 Y 中第一对不同的边（见图 2.9）］。



(a) X^*中的边e_2朝向v　　(b) X^*中的边e_2背向v

图 2.9　定理 2.2 中的两种情形

由于 X^* 和 Y 均是交错路径，边 e_1 和 e_2 有相同的颜色。因为 X^* 是欧拉路径，X^* 包含边 e_2。显然，e_2 跟在 X^* 中的 e_1 后面。

根据边 e_2 在路径 X^* 中的方向，存在两种情形（朝向或背向顶点 v）（见图 2.9）。

情形 1　在路径 X^* 中，边 $e_2=(y_{l+1},v)$ 指向 v。在这种情形，$X^*=x_1\cdots vx_{l+1}\cdots y_{l+1}v\cdots x_m$，因为边 e_1 和 e_2 的颜色一致，变换 $X^*=F_1F_2F_3\longrightarrow F_1\overline{F_2}F_3=X^{**}$ 是一个顺序反射（见图 2.10）。故 $X^{**}\in\mathcal{C}$，且 X^{**} 和 Y 中至少有 $(l+1)$ 个原始顶点是一致的，这与 X^* 的选择矛盾。

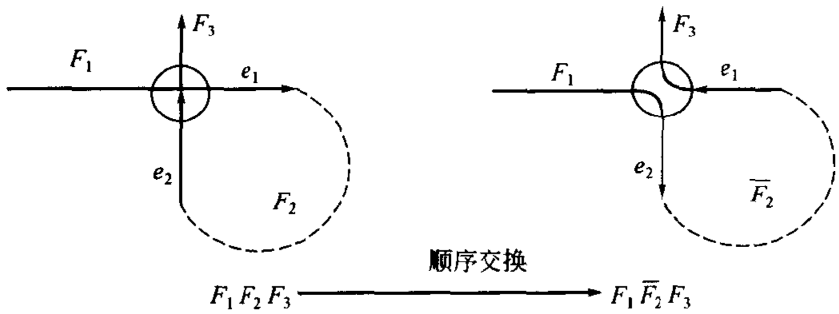


图 2.10　情形 1（顺序交换）

情形 2　在路径 X^* 中，边 $e_2=(v,y_{l+1})$ 背向 v。在这种情形下，顶点 v 把路径 X^* 分拆成 3 条路径，在 v 处结束的前缀 X_1、圈 X_2 和从 v 处出发的后缀 X_3。很容易看出，X_2 和 X_3 有一个公共顶点 $x_j=x_k$（$l<j<k<m$）（否则，Y 就不是一个欧拉圈了）。因此，圈 X^* 可改写成 $X^*=F_1F_2F_3F_4F_5$（见图 2.11）。

考虑图 2.11 中用粗线表示的边 (x_k,x_{k+1}) 和 (x_{j-1},x_j)。如果这些边的颜色不同，那么 $X^{**}=F_1F_4F_3F_2F_5$ 就是从 X^* 中如图 2.11（a）所示的顺序交

换得到的交错圈。X^{**} 和 Y 中至少有（$l+1$）个原始顶点是一致的，这与 X^{*} 的选择矛盾。

如果边（x_k，x_{k+1}）和（x_{j-1}，x_j）的颜色相同［见图 2.11（b)］，那么 $X^{**}=F_1F_4F_2^{-}F_3^{-}F_5$ 可经两个顺序反射 g 和 h 由 X^{*} 获得

$$F_1F_2F_3F_4F_5 \xrightarrow{g} F_1F_2(F_3F_4)^{-}F_5=F_1F_2F_4^{-}F_3^{-}F_5 \xrightarrow{h}$$

$$F_1(F_2F_4^{-})^{-}F_3^{-}F_5=F_1F_4^{--}F_2^{-}F_3^{-}F_5=F_1F_4F_2^{-}F_3^{-}F_5$$

X^{**} 和 Y 中至少有（$l+1$）个原始顶点是一致的，这与 X^{*} 的选择矛盾。■

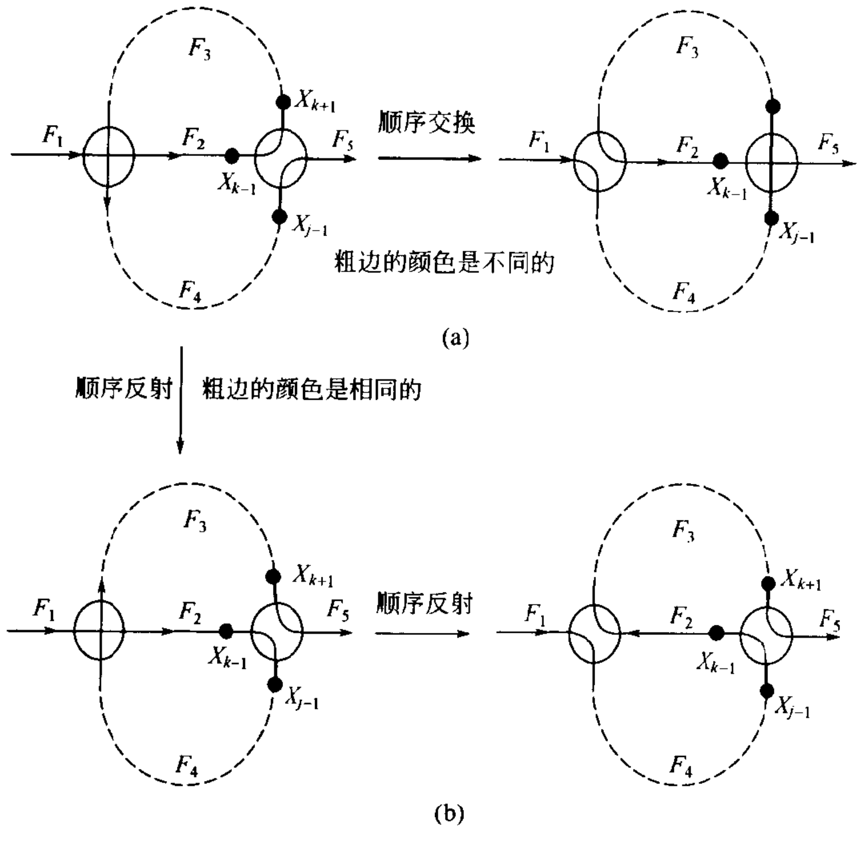


图 2.11　情形 2

根据粗线边的颜色，或存在一个顺序交换或存在两个顺序反射将 X^{*} 变换为具有 Y 最长公共前缀的圈

2.6　物理图谱与交错欧拉圈

这一节中介绍物理图谱的分岔图（fork graphs)，并证明在分岔图中每一个物理图谱与一个交错欧拉路径相对应。

考虑由单酶切 A 和 B，以及双酶切 $C=A+B$：$\{A_1, \cdots, A_n\}$，$\{B_1, \cdots, B_m\}$，$\{C_1, \cdots, C_l\}$ 的顺序片段给出的一个物理图谱。下面为了简单起见，假设 A 和 B 不在同一个位置切割 DNA，即 $l=n+m-1$。片段 A_i 的分岔是包含在

A_i 中的双酶切片段 C_j 的集合

$$F(A_i)=\{C_j: C_j \subset A_i\}$$

(B_i 的分岔可类似地定义)。例如,$F(A_3)$由两个大小为 4 和 1 的片段 C_5 和 C_6 组成(见图 2.12)。显然,每两个分岔 $F(A_i)$和 $F(B_j)$最多有一个公共片段。至少包含两个片段的分岔称为多重分岔。

多重分岔的最左边和最右边的片段称为边界片段。显然,C_1 和 C_l 是边界片段。

引理 2.2 每一个边界片段,包括 C_1 和 C_l,确切地属于两个多重分岔 $F(A_i)$和 $F(B_j)$。边界片段 C_1 和 C_l 确切地属于一个多重分叉。

引理 2.2 激发了具有边界片段长度的顶点集的分岔图的构建。(有相同长度的两个边界片段对应同一个顶点)。分岔图的边集合与所有多重分岔相对应(每个多重分岔由连接顶点的边来表示,而顶点则与它的边界片段的长度相对应)。着色边对应于具有颜色 A 的多重分岔 A,边对应于具有颜色 B 的多重分岔 B(见图 2.12)。

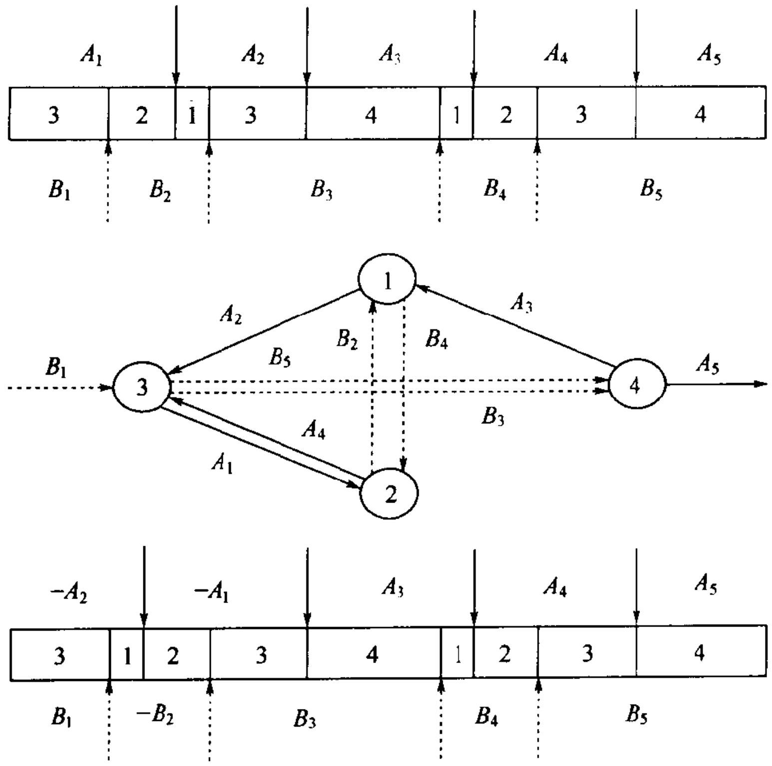


图 2.12 增加了附加边 B_1 和 A_5 的物理图谱的分岔图

实(虚)边对应于多重分岔 A (B)。对应于上面的图谱,这个(无向)图的边上的箭头所指路径为 $B_1A_1B_2A_2B_3A_3B_4A_4B_5A_5$。下面的图谱 $B_1-A_2-B_2-A_1B_3A_3B_4A_4B_5A_5$ 通过改变三角形 A_1, B_2, A_2 中边的方向得到(盒式反射)

图 G 的所有顶点是平衡的，除了可能的顶点 $|C_1|$ 和 $|C_l|$，它们是半平衡的，即 $|d_A(|C_1|)-d_B(|C_1|)|=|d_A(|C_l|)-d_B(|C_l|)|=1$。图 G 可以通过加一条或两条边而转化为平衡图。因此 G 包含了一条交错欧拉圈。

每个物理图谱（A，B）在它的分岔图中都定义了一个交错欧拉路径。物理图谱的盒式变换不改变该图谱的分岔集。产生的问题是具有相同分岔集的两个图谱是否可以用盒式变换进行相互转化。图 2.12 给出了两个具有相同分岔集的图谱，在分岔图中它们对应于两个交错欧拉圈。很容易看到物理图谱的盒式变换对应于分岔图中的顺序变换。因此，在（A，B）的分岔图中的每一条交错欧拉路径对应于由（A，B）通过盒式变换而得到的一个图谱（见定理 2.2）。

2.7 部分酶切问题

部分酶切问题是要在所有位点对之间的 $\binom{n}{2}$ 个距离的集合中重建 n 个限制性酶切位点的位置。如果 ΔX 是 X 中所有点对间距离的（多重）集合，则 PDP 问题就是根据给出的 ΔX 重构 X。Rosenblatt 和 Seymour[289] 给出了用多项式的因式分解求解这个问题的伪多项式算法。Skiena 等[314] 描述了如下简单的向后追踪算法，Skiena 和 Sundaram[315] 则根据带有误差的数据对算法做了进一步的修改。

首先找出 ΔX 中的最长距离，确定 X 的最外面的两个点，然后从 ΔX 中除去这个距离。接着重新选出 ΔX 中剩下距离中最长的。由于每一步中，ΔX 的最长距离必须由最外面的点中的一个来确定，因此只有两种可能位置（左边或右边）放置这个点。在每一步中，分别对两个位置中的每一个检查从该位置到所有已选点的距离是否在 ΔX 中。如果是，在进行下一步之前除去所有这样的距离。如果这两个位置都不满足，就进行向后追踪。当 ΔX 变为空时，就得到了一个解。

例如，假设 $\Delta X=\{2,2,3,3,4,5,6,7,8,10\}$。因为 ΔX 包含了所有点对间的距离，那么 $|\Delta X|=\binom{n}{2}$，其中 n 是解中位点的数目。首先设 $L=\Delta X$，$x_1=0$。因为 10 是 L 中的最大距离，很清楚 $x_5=10$。从 L 中除去距离 $x_5-x_1=10$，得到

$$X=\{0,10\},\quad L=\{2,2,3,3,4,5,6,7,8\}$$

剩下的距离中最大的为 8。现在有两种选择：$x_4=8$ 或 $x_2=2$。因为这两种情况互为镜面映照，不失一般性，可以假设 $x_2=2$。在从 L 中除去距离 $x_5-x_2=8$ 和 $x_2-x_1=2$ 后，得到

$$X=\{0,2,10\},\quad L=\{2,3,3,4,5,6,7\}$$

因为 7 是剩下的最大距离，有 $x_4=7$ 或 $x_3=3$。如果 $x_3=3$，则距离 $x_3-x_2=1$ 必须在 L 中，但它不在，所以只能设 $x_4=7$。在 L 中除去了距离 $x_5-x_4=3$，$x_4-x_2=5$ 和 $x_4-x_1=7$ 后，得到

$$X=\{0,2,7,10\},\quad L=\{2,3,4,6\}$$

现在 6 是剩下的最长距离了。再次有两种选择：$x_3=4$ 或 $x_3=6$。如果 $x_3=6$，则距离 $x_4-x_3=1$ 必须在 L 中，但它不在。所以只剩下 $x_3=4$，得到部分酶切问题的一个解 {0，2，4，7，10}。

所述算法的伪代码在下面给出。函数 **Delete _ Max**（L）返回 L 的最大值，且从 L 的列表中除去这个值，函数定义了两个全局变量 X 和 *width*。$\Delta(X, Y)$ 是所有 X 的点和 Y 的点间距离的（多重）集合。

```
设置 X
int width
Partial_Digest(List L)
    width=Delete_Max(L)
    X={0,width}
    Place(L)
Place(Lis L)
    If L=∅ then
      output solution X
      exit
    y=Delete_Max(L)
    if Δ({y},X)⊂L then
      X=X∪{y}
        Place[L\Δ({y},X)]                /* 在右边位置放一个点 */
        X=X\{y}                          /* 向后追踪 */
    if Δ({width-y},X)⊂L then
        X=X∪{width-y}
        Place[L\Δ({width-y},X)]          /* 在左边位置放一个点 */
        X=X\{width-y}                    /* 向后追踪 */
```

如果 L 在通常位置上是由实数点产生的，那么这个算法运行的期望时间为 $O(n^2\lg n)$，因为在每一步中，两个选择中的一个将以概率 1 被除去。然而，算法的运行时间在最差的情况下可能会呈指数增长[377]。

2.8 同效集

从 ΔX 中不可能总是惟一地重构出一个点集 X。集合 A 和 B 是同效的，如

果 $\Delta A=\Delta B$。令 U 和 V 是两个多重集合。容易证明多重集 $U+V=\{u+v: u\in U, v\in V\}$ 和 $U-V=\{u-v: u\in U, v\in V\}$ 是同效的。图 2.13 中的例子是从 $U=\{6, 7, 9\}$ 和 $V=\{-6, 2, 6\}$ 的结构中产生的。

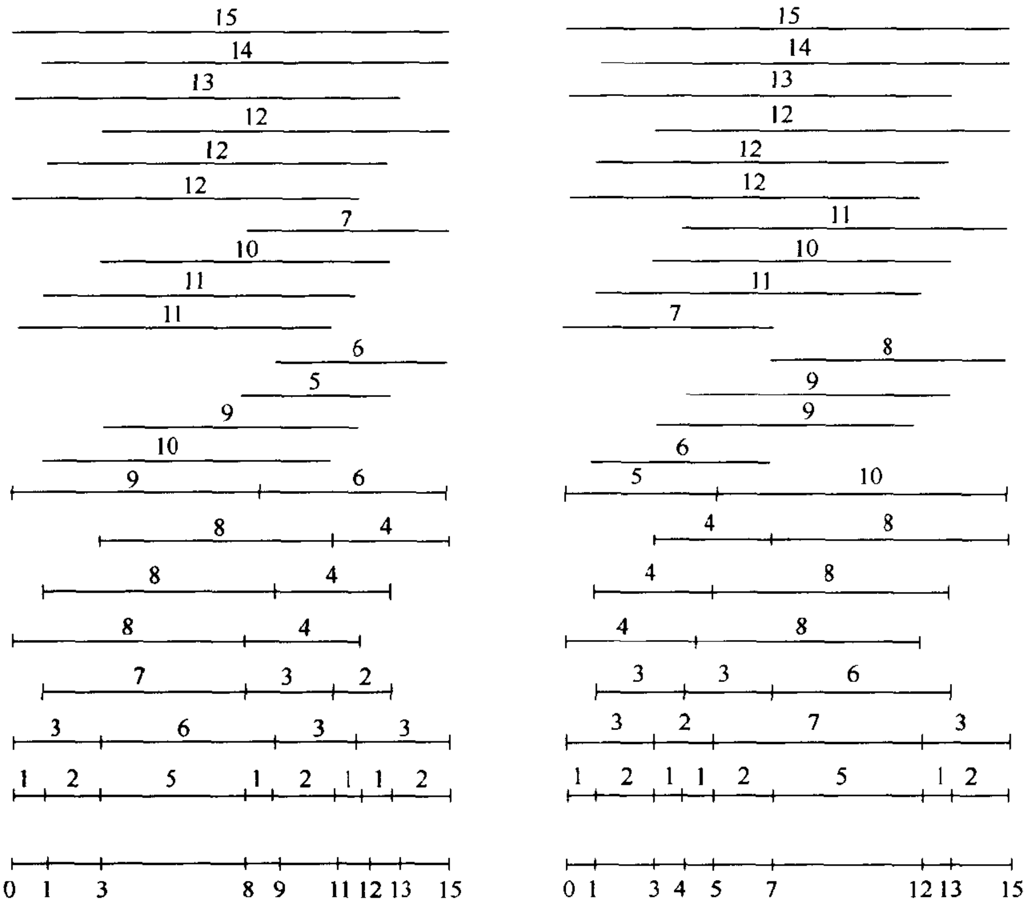


图 2.13 同效集 $U+V=\{0, 1, 3, 8, 9, 11, 12, 13, 15\}$ 和 $U-V=\{0, 1, 3, 4, 5, 7, 12, 13, 15\}$

自然会问每一对同效集是否表示了这个结构的一个实例。回答是否定的：同效集 {0，1，2，5，7，9，12} 和 {0，1，5，7，8，10，12} 提供了一个反例。然而，如果定义 U 和 V 是“可能负重数的多重集”(multisets with possibly negative multipcities)，那么这种猜测就是正确的。

给定一个整数多重集 $A=\{a_i\}$，令 $A(x)=\sum_i x^{a_i}$ 是 A 的母函数。容易看到 ΔA 的母函数是 $\Delta A(x)=A(x)A(x^{-1})$。令 $A(x)$ 和 $B(x)$ 是多重集 A 和 B 的母函数，使得 $A(x)=U(x)V(x)$，$B(x)=U(x)V(x^{-1})$。那么，$A(x)A(x^{-1})=B(x)B(x^{-1})=U(x)V(x)U(x^{-1})V(x^{-1})$，隐含 A 和 B 是同效的。

定理 2.3[289] 两个集合 A 和 B 是同效的，当且仅当存在母函数 $U(x)$ 和 $V(x)$ 和一个整数 β，使得 $A(x)=U(x)V(x)$，$B(x)=\pm x^{\beta}U(x)V(x^{-1})$。

证明 令 A 和 B 是同效集。令 $P(x)$ 是 $A(x)$ 和 $B(x)$ 的最大公约数，又令

$A(x)=P(x)Q_A(x)$，$B(x)=P(x)Q_B(x)$，其中 $Q_A(x)$ 和 $Q_B(x)$ 是互质的。令 $V(x)$ 是 $Q_A(x)$ 和 $Q_B(x^{-1})$ 的最大公约数，又令 $Q_A(x)=V(x)S_A(x)$，$Q_B(x^{-1})=V(x)S_B(x)$，其中 $S_A(x)$ 和 $S_B(x)$ 是互质的。显然，$S_A(x)$ 和 $S_A(x^{-1})$ 对 $S_B(x)$ 和 $S_B(x^{-1})$ 均是互质的。

因为 A 和 B 是同效的，于是

$$P(x)V(x)S_A(x)P(x^{-1})V(x^{-1})S_A(x^{-1})$$
$$=P(x)V(x^{-1})S_B(x^{-1})P(x^{-1})V(x)S_B(x)$$

隐含 $S_A(x)S_A(x^{-1})=S_B(x)S_B(x^{-1})$。因为 $S_A(x)$ 和 $S_A(x^{-1})$ 对 $S_B(x)$ 和 $S_B(x^{-1})$ 均是互质的，$S_A(x)=\pm x^a$ 和 $S_B(x)=\pm x^b$。因此，$A(x)=\pm x^aP(x)V(x)$，而 $B(x)=\pm x^bP(x)V(x^{-1})$。代入 $U(x)=\pm x^aP(x)$，定理得证。 ■

会发生这样的情况，定理 2.3 中给出的一些分解系数是负的，对应于有负重数的多重集。例如，如果 $A=\{0,1,2,5,7,9,12\}$，$B=\{0,1,5,7,8,10,12\}$，$U(x)=\{1+x+x^2+x^3+x^4+x^5+x^7\}$，$V(x)=x^{-5}(1-x^3+x^5)$。

集合 A 是可重构的，如果任何时候 B 与 A 是同效的。对数 v，有 $B=A+\{v\}$，或 $B=-A+\{v\}$。集合 A 称为是对称的，如果对某一数 v，有 $-A=A+v$。多项式 $A(x)$ 是对称的，如果相应的集合是对称的，即 $A(x^{-1})=x^vA(x)$。定理 2.3 隐含下面的定理。

定理 2.4 集合 A 是可重构的，当且仅当 $A(x)$ 至多有一个非对称的素因子（可计重数）。

Rosenblatt 和 Seymour[289] 给出了下面的有 n 个点的部分酶切问题的伪多项式算法。给定 $\binom{n}{2}$ 个距离的集合 $\Delta A=\{d_i\}$，构造母函数 $\Delta A(x)=n+\sum_i(x^{d_i}+x^{-d_i})$。可把这个多项式通过整系数多项式环分解为不可约，用一个运行时间为 $\max_i d_i$ 的多项式的因式分解算法。PDP 问题的解 $A(x)$ 一定有形式 $\Delta A(x)=A(x)A(x^{-1})$。因此，尝试所有 $\Delta A(x)$ 的 F 个不可约非互易因子的 2^F 个可能子集 S 为 $A(x)$ 的拟定因子。作为结果，找到所有可能集合 $A(x)$ 的一个集合。最后，除去有负系数的集合 $A(x)$，并分类删除可能多余的拷贝。

2.9 若干其他的问题和方法

2.9.1 光学作图

Schwartz 等[311]利用光学作图技术进行限制性酶切图谱的构建。在光学作图中，DNA 分子的单拷贝被伸展并贴在显微镜下的玻璃上。当限制酶被激活后，它们便在限制位点上切割 DNA。分子仍然贴在表面上，但伸展的 DNA 分子的弹性把分子的末端拉回切割点。在显微镜下，这像分子荧光线上的微小裂缝而被

识别。于是，切割位点处有裂缝的 DNA 分子“照片”给出了限制性酶切图谱的快照。

光学作图解决了重构限制片段的顺序问题，但产生了新的计算上的挑战。问题是，每个分子中并不是所有的限制位点都被切断了（假阴性），而有一些非限制位点却又可能不正确地被切割（假阳性）。另外，片段长度测定的不精确性、分析邻近限制位点的困难以及每个分子的定向未知（从左到右或反过来）使得重构变得困难。实际上，许多分子的数据被收集来建立一个共有的限制酶切图谱。

方向未知的问题由 Muthukrishman 和 Parida[243] 归结为二进制翻转切割（Binary Flip-Cut，BFC）问题。在 BFC 问题中，给出由 n 个二进制 0-1 字符串组成的集合（每个字符串表示一个 DNA 分子的快照，其中 1 对应于一个限制位点）。问题是对每一个串指定翻转或不翻转状态，使得共有限制位点数目最小。一个位点称为在翻转指定下是共有的，如果对某一小的定常参数 c，在分子翻转的情况下，至少有 cn 个 1 在那个点出现。

处理真实的光学作图数据被认为是较之 BFC 问题更为困难。光学作图问题的有效算法由 Anantharaman 等[8]，Karp 和 Shamir[190] 以及 Lee 等[218] 发展。

2.9.2 加探针部分酶切作图

导出物理图谱的另一个技术是加探针部分酶切问题（probed partial digest problem，PPDP)。这种方法中，DNA 被一个限制性酶部分酶切，那样产生任意两个切割位点间的 DNA 片段的一个集合。然后一个有标签的探针附着在两个切割位点间的 DNA 上，与部分酶切的 DNA 进行杂交，测量与探针杂交的片段的大小。问题是从测得的长度多重集中重建酶切位点所处的位置。

在 PPDP 问题中，多重集 $X\subset[-s,t]$ 被分拆为两个子集 $X=A\cup B$，其中 $A\subset[-s,0],B\subset[0,t]$，对应于探针从限制性位点向左和向右。PPDP 实验提供了多重集 $E=\{b-a:a\in A,b\in B\}$。问题是要在给定的 E 中找到 X。Newberg 和 Naor[252] 指出 PPDP 解的数目会很快增长，至少比 $n^{1.72}$ 快。

3 图谱装配

3.1 引言

图谱装配问题可以根据下述类比来理解。想像一下一本书的几个拷贝被剪刀剪切为成千上万块碎片。每份拷贝是由独立的方法来剪的，所以一份拷贝的碎片有可能和其他拷贝的碎片交叠。对每块碎片和一系列关键词中的每个词，要问碎片是否包含关键词。对给出的数据，希望决定碎片交叠的模式。

双酶切和部分酶切技术可以使生物学家构建小的 DNA 分子限制酶切（物理）图谱，例如病毒、叶绿体或线粒体 DNA。然而这些方法不能（实验上或计算上）应用于大的 DNA 分子。尽管病毒基因组的第一张限制酶切图谱是 1973 年建成的，但是，通过小片段的限制酶切图谱的装配来构建第一张细菌基因组的物理图谱却花费了十多年的时间。为了研究大的 DNA 分子，生物学家先把它断裂为较小的片段，再建立每个片段的酶切图谱或指纹图谱，然后通过装配片段来决定整个分子的图谱。1986 年，这种作图战略最早是由 Olson 等[257]为构建酵母的限制酶切图谱和 Coulson 等[76]为构建线虫的限制酶切图谱而发展起来的。然而，第一张大尺度物理图谱则是由 Kohara 等在 1987 年[204]构建的大肠杆菌物理图谱。

作图通常开始于用限制酶将 DNA 分子切割成许多小的片段。为了研究单个片段，生物学家通过对它们的克隆得到许多相同的片段。克隆将 DNA 片段插入克隆载体，它是一个小的、来源于病毒或其他生物体的人工构建的 DNA 分子。有 DNA 插入的克隆载体被导入到细菌的自我复制宿主。自我复制过程产生大量的片段拷贝，从而使它的结构可以被系统地检测。DNA 片段以这种方式繁殖被称为克隆。

作为结果，生物学家们获得了一个克隆文库，它包含成千上万个来自同一 DNA 分子的克隆（每个代表了一个短的 DNA 片段）。克隆文库中的克隆可能交叠（交叠可以通过使用一些不同的限制酶来实现）。在克隆文库建成后，生物学家希望排列克隆，即沿着 DNA 分子重建克隆的相对位置。在克隆文库的建造过程中信息可能丢失，因此重建过程从构建克隆的指纹图谱开始。其思想是使用一个易于确定的指纹图谱来标记每个克隆，而指纹图谱则可看作一套出现在一个克隆中的“关键词”。如果两个克隆基本上交叠，那么它们的指纹图谱应该是相似的。而不交叠的克隆不可能有相似的指纹图谱，于是这些指纹图谱可以使生物学

家区分交叠和非交叠的克隆，并重构克隆的顺序。下述指纹图谱被用于许多作图计划。

• 限制酶切图谱　克隆的限制酶切图谱提供一列有序的限制片段。如果两个克隆的限制酶切图谱共享若干连续片段，它们可能是交叠的。运用这个战略，Kohara 等[204]构建了大肠杆菌基因组的物理图谱。

• 限制酶切片段的大小　限制酶切片段的大小可通过用限制酶切割克隆并测量所得片段的大小来得到。这比构建限制酶切图谱简单。尽管无序的片段大小排列包含的信息比有序的排列少，但它仍然提供一个适当的指纹图谱。这种类型的指纹图谱被 Olson 等[257]用于酵母作图计划。

• 杂交数据　在这种方法中，一个克隆暴露在大量的探针中，并决定哪些探针与克隆杂交。探针可以是短的随机序列或是任何在实验中已鉴定的 DNA 片段。一个特别有用的探针类型是序列标签位置（Sequence Tag Site，STS）。STSs 从 DNA 链自身提取，且经常从克隆的末端提取。每个 STS 是足够长的，这使它不可能在 DNA 链中二次出现；因此，它确定了 DNA 链上的惟一位置。利用 STS 作图，Chumakov 等[68]和 Foote 等[111]构建了第一张人类基因组的物理图谱。

STS 技术导致具有惟一探针的作图。如果探针是短的随机序列，那么它们可能与 DNA 在很多位置杂交，于是就导致非惟一探针的作图。对具有 n 个克隆和 m 个探针的图谱装配问题，其实验数据是一个 $n\times m$ 的矩阵 $D=(d_{ij})$，其中 $d_{ij}=1$，如果克隆 C_i 覆盖探针 p_j，而其他情况 $d_{ij}=0$（见图 1.1）。注意，数据并不表明一个探针在给定的克隆上出现了多少次，也不表明探针在克隆上的顺序。如果存在 S 的一个子字符串严格覆盖与 C 相同的探针集（在子字符串中忽略探针的顺序和多样性），一个字符串 S 覆盖一个克隆 C。图 1.1 中的字符串覆盖了相应于杂交矩阵 D 的 9 个克隆中的每一个。最短覆盖字符串问题就是在探针的字母表中找出包含所有克隆的最短字符串。

最短覆盖字符串问题是 NP 完全的。然后，如果克隆的顺序是固定的话，它可以在多项式时间内求解。Alizadeh 等[3]提出了求解最短覆盖字符串问题的局部改进战略，它是在克隆顺序固定的情形下寻找最优交错。

在直线上给定一组区间，可以生成该集合的区间图，通过将图的顶点与每个区间连接起来，并且，如果相应的区间交叠则可用一条边连接两顶点（见图 3.2）。在惟一探针的情形，每个无误差的杂交矩阵定义一个克隆顶点集合的区间图，其中如果克隆 i 和克隆 j 有共用探针的话，则由边将它们连接起来。区间图的研究最初是由 Benzer 开始的，他得到噬菌体 T4 DNA 的成对片段间的交叠数据。他在处理交叠数据方面获得成功，在某种程度上这意味着基因的线性特征。Benzer 的问题可以阐述如下：给定基因组的两个片段是否交叠的信息，这数据

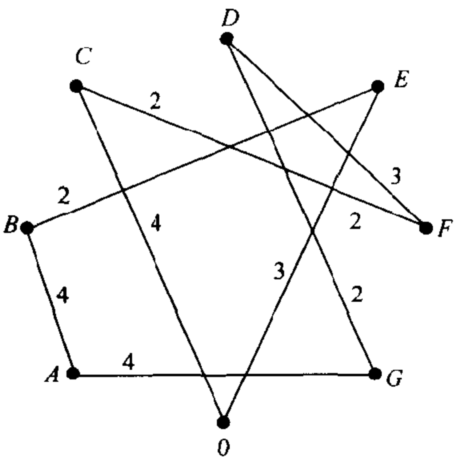
与基因是以线性顺序排列的假设一致吗？这个问题与交叠图是不是区间图的问题等价。

区间图与具有相邻 1 性质（consecutive ones property）的矩阵密切相关。一个（0，1）矩阵有相邻 1 性质，如果它的列可以用出现在相邻位置的每一行的 1 来置换。在惟一探针情形，每个无误差的杂交矩阵有相邻 1 性质（这要求列的置换与探针从左到右顺序排列相符）。给定一个任意矩阵，对可以检测它是否有相邻 1 性质的算法感兴趣。区间图的特征和有相邻 1 性质的矩阵是由 Gilmore 和 Hoffman[128] 和 Fulkerson 和 Gross[114] 给出的。Booth 和 Leuker[40] 发展了称为 PQ 树（PQ-tree）的数据结构，它导致一个鉴别相邻 1 性质的线性时间算法。给定一个无误差的杂交矩阵，Booth-Leuker 算法在线性时间内对探针的所有正确排列构建了一个紧凑的表示。然而，他们的方法无法忍受实验误差。Alizadeh 等[3] 为探针的排序设计了一个选择性的简单程序。在有实验错误的情形程序可能会失败，然而在大多数情形下它对探针顺序结构仍有很好的启发。

杂交误差的最普通类型是假阴性（false negative），其影响是误差虽在探针和克隆之间发生但却未被察觉。另外，杂交数据还受假阳性（false positive）以及由于克隆异常而产生的误差的影响。不同的克隆技术产生不同的克隆异常。每个克隆文库都是建立在噬菌体 λ 载体上的，且最多容纳 25kb 的 DNA。黏粒（cosmid）代表另一种载体，它把来自质粒（plasmid）的 DNA 序列结合到 λ 基因组的某个区域。使用黏粒，最长的插入 DNA 为 45kb，故用 70000 个黏粒即可覆盖整个人类基因组。为了减少载体的数量，酵母人工染色体（yeast artificial chromosome，YAC）被发展出来用于克隆更长的 DNA 片段（最长可达 1000kb）。尽管酵母人工染色体已被用于许多作图计划，但仍有许多异常克隆与其结合。最常见的异常是嵌合（chimerism）现象。一个嵌合的克隆包含两个不同的 DNA 片段，它们在克隆过程中被错误地连接在一起。在 YAC 文库中估计有 10%～60%的克隆是嵌合的。嵌合现象可能起因于不同 DNA 片段的互绑或两个 DNA 分子的重组。使用酵母人工染色体的另一个问题是，许多克隆是不稳定的，且易删除内部区域。与酵母人工染色体相比，基于大肠杆菌基因组的细菌人工染色体（Bacterial Artificial Chromosome，BAC）克隆系统明显地减少了嵌合现象。

探针的真实排序相应于杂交矩阵 D 的列的置换 π，它产生矩阵 D_π。D_π 的每行相应于含有单个 1 块的正常克隆，而相应于嵌合克隆的每行则有两个 1 块。在行中定义空缺为零块，它们是侧面相接的。在 D_π 中，空缺的数量等于嵌合克隆的数量。典型的假阴性误差是将一个 1 块分裂为两个部分，因此产生了空缺。一个典型的假阳性误差是将一个空缺分裂为两个空缺。因此在 D_π 中空缺的数目将近似等于嵌合克隆数目加杂交误差的数目。这提示了一种启发式的方法，将空缺

数变得最小的列的变换将等同于好的探针排序[3,137]。空缺数的最小化可归结为称作汉明距离（Hamming distance）的流动销售员问题（Traveling Salesman Problem，TSP），其中城市由 D 的列和附加的所有元素为零的列表示，两个城市间的距离为相应列间的加权平均距离，即两列中的城市位置标记数是不同的（见图 3.1）。



探针

克隆	C	F	D	G	A	B	E
1	1				1		
2				1	1	1	1
3	1	1	1	1		1	1
4	1	1	1	1		1	
5	1	1	1	1			
6		1	1	1	1		
7			1	1	1	1	1
8			1	1	1	1	
9			1				

图 3.1 加权平均距离 TSP 的最短闭链

其相应于图 1.1 中具有列不同次序的杂交矩阵，所示的

最小闭链定义具有最小空缺数的克隆的次序。

在此排序中，克隆 1，3 和 4 是嵌合的

3.2 非惟一探针的作图

使用短探针杂交指纹图谱的物理作图是由 Poustka 等[279]提出的。这种方法的优点在于探针的合成便宜又简单。但是，使用短探针的杂交实验误差率很高，其结果是，用非惟一探针很少有成功的作图计划[165]。杂交误差以及缺乏构建图谱的好算法成了在大尺度作图计划中使用这种方法的主要障碍。最近，Alizadeh 等[3]及 Mayraz 和 Shamir[234]设计了一种算法，它在出现杂交误差时仍能很好地工作。

在线［0，N］上给每个克隆分配一个区间称为一个布局（placement）（线［0，N］相当于整个 DNA 分子）。交叉是指这 n 个区间的 $2n$ 个端点的线性排列的一种规范。一个交叉可以看成具有共同拓扑结构的布局的等价类。给定一个矩阵 D，图谱装配问题是决定最大可能的交叠[3]。

Alizadeh 等[3]给出了对 Lander 和 Waterman[214]随机模型的最大可能性概念的精确理解，并说明它相应于最短包含序列。在这个模型中，相同长度的克隆被相互独立地放置在 DNA 的随机位置上，而 DNA 上的探针则根据相互独立的泊

松过程来定位。Alizadeh 等[3]设计了这种模型的最大拟然函数。这函数的最简单近似是寻找一个交叉，使得解释杂交数据所需的探针出现的次数为最小，这就是最短覆盖字符串问题。

在以下讨论中，假设没有克隆完全地包含其他的克隆。序列 S 包含克隆的一个排列 π，如果序列以 π 给定的顺序包含所有的克隆。一个序列 $ABACBACDBCE$ 包含克隆 C_1、C_2、C_3、C_4 的排列（3，2，4，1），它同下列探针 A、B、C、D、E 杂交

$$C_1-\{B,C,E\},\quad C_2-\{A,B,C,D\},\quad C_3-\{A,B,C\},\quad C_4-\{B,C,D\}$$

设 $c(\pi)$ 为覆盖 π 的最短字符串的长度。图 1.1 代表了对应长度为 $c(\pi)=15$ 的排列 $\pi=$（1，2，…，9）的最短覆盖字符串，Alizadeh 等在 1995 年[3]设计了一种多项式算法用来寻找 $c(\pi)$，以及一种求近似 $\min\limits_{\pi} c(\pi)$ 的局部改进算法。

克隆的端点将线段分成许多原子区间（atomic interval），其中每个原子区间是不包含克隆端点的最大区间（见图 3.2）。每个原子区间包含在克隆 i，…，j 中，其中 $i(j)$ 代表包含该原子区间最左边（最右边）的克隆，则定义这样的原子区间为 $[i, j]$。如果 $i<i'<j'<j$，区间 $[i, j]$ 和 $[i', j']$ 是相互冲突的。注意到每组原子区间是不冲突的，即不包含冲突的区间（再次假设没有克隆完全包含另一个克隆）。

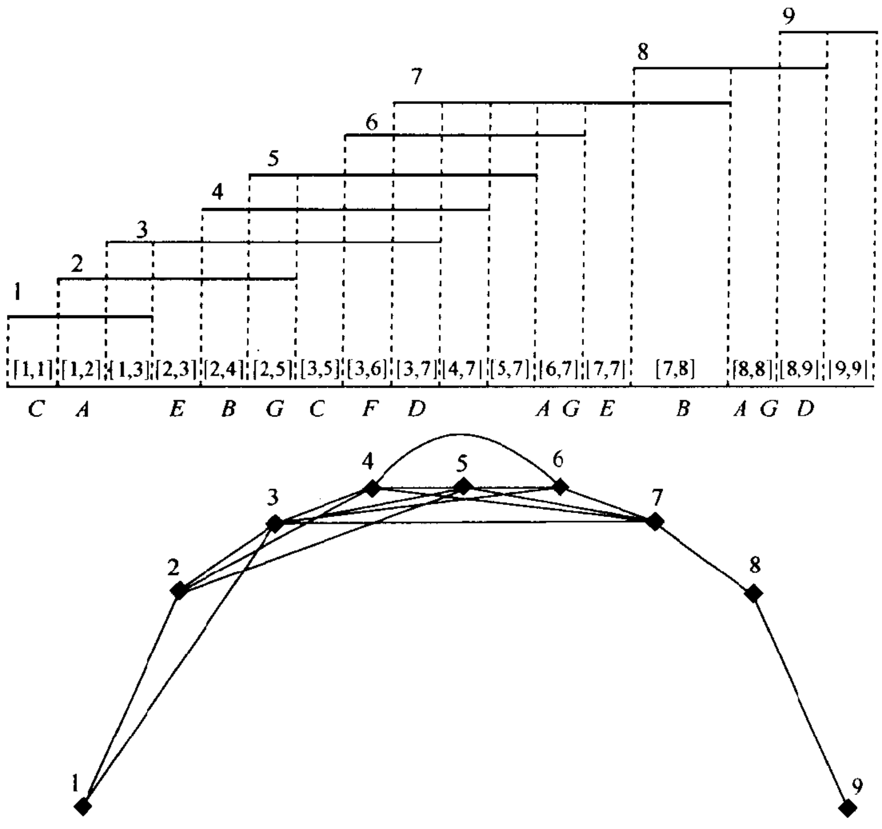


图 3.2 原子区间

一组区间 $\mathcal{I}=([i,j])$ 被称为是相容的，如果存在一个克隆的交叉使得 $\mathcal{I}$ 中的每个区间都是原子区间。

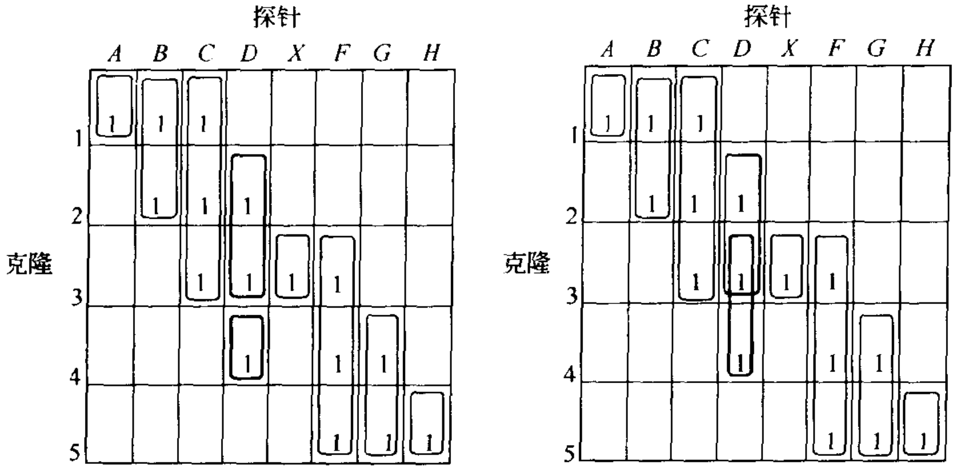
引理 3.1 一组区间是相容的当且仅当它是不冲突的。

证明 归纳为克隆数。

对图 1.1 中的数据，包含探针 A 的克隆由两区段组成：[1，2] 和 [6，8]。在一个覆盖字符串中的每个探针 A 在相应的克隆交叉中定义一个原子区间 $[i, j]$，并因此在杂交矩阵的 A 列生成一个长度为 $j-i+1$ 的区段。因此探针 A 在每个覆盖字符串中至少出现两次。它给出了对最短覆盖字符串长度的下界 $2+2+2+1+2+1+1=11$。然而并没有长度为 11 的覆盖字符串，因为在杂交矩阵中部分区段是冲突的（例如 C 的 [3，5] 区段同 G 的 [2，8] 区段冲突）。注意到每个长度为 t 的覆盖字符串定义一组相容的 t 区段（区间）。另一方面，每个相容的 t 区段组定义一个长度为 t 的覆盖字符串。这个观察资料和引理 3.1 意味着，最短覆盖字符串问题可以用如下形式表示：杂交矩阵的一个区段组可再拆分为有最少区段数的无冲突集合。例如，为了避免区段 [3，5] 同区段 [2，8] 间的冲突，可以将区段 [2，8] 再拆分为 [2，5] 和 [6，8]。这种再拆分仍留下 E 的区间 [7，7] 同 A 的区间 [6，8] 冲突，D 的区间 [3，9] 和 G 的区间 [6，8] 冲突。将这三个区间再拆分后，最终可得包含 $11+4=15$ 个区段的无冲突集合（见图 1.1 中右边的矩阵）。这 15 个区段生成最短覆盖字符串

$$\underbrace{[1,1]}_{C}\underbrace{[1,2]}_{A}\underbrace{[2,3]}_{E}\underbrace{[2,4]}_{B}\underbrace{[2,5]}_{G}\underbrace{[3,5]}_{C}\underbrace{[3,6]}_{F}\underbrace{[3,7]}_{D}\underbrace{[6,7]}_{A}\underbrace{[6,7]}_{G}\underbrace{[7,7]}_{E}\underbrace{[7,8]}_{B}\underbrace{[8,8]}_{A}\underbrace{[8,8]}_{G}\underbrace{[8,9]}_{D}$$

下面叙述的是再拆分的贪婪算法，它用来产生最短覆盖字符串。设 $[i, j]$ 和 $[i', j']$ 是两个冲突的区段，其中 $i<i'\leqslant j'<j$，且 j' 是所有冲突区段中最小的。显然，在每次再拆分中 $[i, j]$ 必须在 j' 之前被切断（即在每个 $[i, j]$ 的再拆分中存在一个区间 $[i, t]$，其中 $t<j'$）。这种观察结果提出了一种基于切割每个区间“尽可能朝后”（as late as possible）以避免冲突的再拆分战略。更准确的是，给出一个区段 $[i, j]$，设 $\mathcal{I}$ 为包含在 $[i, j]$ 中的区间的集合。设 t 为 $\mathcal{I}$ 中相互不交叠的区间的最大数目，那么 $[i_1, j_1]$ 是 $\mathcal{I}$ 中有最小的 j_1 的区间，$[i_2, j_2]$ 是 $\mathcal{I}$ 中所有 $i_2>j_1$ 的区间中具有最小 j_2 的区间，…，$[i_t, j_t]$ 则为 $\mathcal{I}$ 中所有 $i_t>j_{t-1}$ 的区间中具有最小 j_t 的区间。初看起来，这像把区段 $[i, j]$ 拆分成 $t+1$ 个区段 $[i, j_1]$，$[j_1+1, j_2]$，$[j_2+1, j_3]$，…，$[j_{t-1}+1, j_t]$，$[j_t+1, j]$ 这样一来便可以解决最短覆盖字符串问题了。但不幸的是，这并不正确，其例子就像在图 3.3 中说明的（由 Tao Jiang 提供）。然而，只要对 $t+1$ 个区间中的最后一个区间进行简单的修改就可以解决这个问题了（见图 3.3）。注意到在这种情形下在最短覆盖字符串中有一个克隆两次出现探针 D。



(a) 无效的子分割
D中的区间[4,4]和
F中的区间[3,5]冲突

(b) 有效的子分割
D中的区间[2,4]拆分为重叠的区间[2,3]
和[3,4]，避免与F中的区间[3,5]冲突

最短覆盖字符串

```
ABC
 BCD
  CDXDF
     DFG
       FGH
```

图 3.3　包含探针重复出现的克隆的最短覆盖字符串（例如 $CDXDF$）

3.3　惟一探针的作图

设 $\{1,\cdots,m\}$ 为一组探针，又设 C_i 为标有探针 i 的克隆的集合。克隆 X 包含克隆 Y，如果标记在 X 上的一组探针（严格地）包含标记在 Y 上的一组探针。下面假设克隆的集合满足如下条件。

- 非包含性　没有包含其他克隆的克隆。
- 连通性　可将任何探针集合拆分成两个非空的集合 A 和 B，存在探针 $i\in A$ 和探针 $j\in B$ 使得 $C_i\cap C_j$ 非空。
- 可区别性　当 $i\neq j$ 时，$C_i\neq C_j$。

这些假设不失一般性，因为任何一组克隆可以通过排除包含在其他克隆中的克隆来得到过滤。下述引理表述了连续克隆的特性。

引理 3.2　设 $\{1,\cdots,m\}$ 为探针的恰当排列，且 $1\leqslant i<j<k\leqslant m$。那么在无误差的情形下，$|C_i\cap C_j|\geqslant|C_i\cap C_k|$，又 $|C_k\cap C_j|\geqslant|C_i\cap C_k|$。

给定一个探针 i，怎样在探针的恰当排列中找到相邻的探针 $i-1$ 和 $i+1$？引理 3.2 提出这些探针应该在使得 $|C_i\cap C_k|$ 最大的探针中——也就是或者 $|C_i\cap$

$C_{i-1}|=\max_{k\neq i}|C_i\cap C_k|$，或者$|C_i\cap C_{i+1}|=\max_{k\neq i}|C_i\cap C_k|$。如果只有一个探针 k 使得$\max_{k\neq i}|C_i\cap C_k|$，那么或者 $k=i-1$，或者 $k=i+1$。如果有一些探针使得$\max_{k\neq i}|C_i\cap C_k|$，则容易看出其中有最小$|C_k|$的探针要么是 $i-1$ 要么是 $i+1$。上述观察可导出一种探针排序的有效算法，它可以从任何一个探针 i 开始并尝试在下一步寻找相邻的探针（$i-1$ 或 $i+1$）。在第（$k+1$）步，利用引理 3.2，该算法可尝试将已经找到的 k 个连续探针的区域向右或左延伸。

对每个探针 i，定义探针集合的偏序$>_i$，$j>_ik$ 如果

$$|C_i\cap C_j|>|C_i\cap C_k|$$

或者

$$|C_i\cap C_j|=|C_i\cap C_k|\neq\varnothing 且 |C_j|<|C_k|$$

显然，如果 $j>_ik$，那么在真实的排列中，探针 k 不可能在探针 i 和探针 j 之间。而且，在$>_i$中最大的元件是 $i-1$ 或者是 $i+1$。

设 $N(i)$ 为探针集合，探针 i 在至少一个克隆上出现。

引理 3.3 满足下列条件的每个 i 惟一决定探针的真实排列。

- $N(i)$中的探针在排列中连续地出现而生成块 $B(i)$。
- 由探针 i 开始向右或向左移动，$B(i)$中的探针在偏序$>_i$中形成一个递减的链。

引理启发了如下算法，它可以找到探针的真实排列[4]。在这算法中，变量$\pi=\pi_{\text{first}}\cdots\pi_{\text{last}}$表示真实排列中连续探针的序列。开始时，$\pi$ 被单个探针的序列初始化。每一步根据如下步骤将一个元件同 π 连接。

- 如果 $N(\pi_{\text{last}})$中每个元件位于 π 中，则将序列 π 用其逆序替换。
- 选择一个探针 $k\notin\pi$，它在 $N(\pi_{\text{last}})$ 的探针中以排列$>_{\pi_{\text{last}}}$为最大。如果$\pi_{\text{last}}>_k\pi_{\text{first}}$，那么将 k 加到 π 的末尾；反之将 k 加到 π 的开头。

当 π 包含所有的探针时，算法停止。引理 3.2 意味着每步之后，π 中的探针形成了同真实排列相同的连续块。因为在每步中有一个新的元件同 π 相连接，所以这种算法要 m 步之后才能给出真实的排列 π。Alizadeh 等[4]报道了对这种算法的简单改进，使得尽管有杂交错误和嵌合克隆的存在也可以很好地运行。

3.4 区间图

Golumbic 在 1980 年[132]极好地介绍了区间图，下面参照这本书进行叙述。

三角剖分图是每个长度大于 3 的简单回路都有弦的图。图 3.4（a）中“房形”图并不是三角剖分图，因为它有一个无弦的长度为 4 的回路。

引理 3.4 每个区间图都可三角剖分。

并不是所有的三角剖分图都是区间图。例如，图 3.4 (b)中的星形图并不是

区间图（证明它!）。一张无向图有可迁定向性（transitive orientation property），如果每条边都可以给出一个方向使得图 $G(V, E)$ 满足下列条件，对每三个顶点 a，b，c：若 $(a,b)\in E$ 且 $(b,c)\in E$，那么 $(a,c)\in E$。一个无向图如果它是可迁定向的，那么它被叫作可比图（comparability graph）（见图 3.4）。图 3.4（b）中星形图的补图并不是一个可比图。

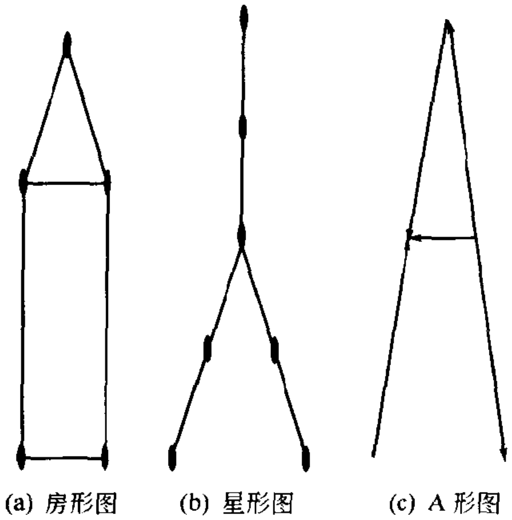


图 3.4　(a) 房形图不是区间图，因为它不可三角剖分。(b) 星形图不是区间图，因为它的补图不是一个可比图。(c) A 形图是可迁定向的

引理 3.5　一个区间图的补图是一个可比图。

证明　对无交叠的区间 $[i,j]$ 和 $[i',j']$，其中 $j\leqslant i'$，在区间图的补图的相应的边上给出方向 $[i,j]\rightarrow[i',j']$。■

一个完全图是指每对顶点之间都有边连接的图。图 G 的一个团（clique）是 G 的完全子图。一个团是最大的，如果它不包含在其他任何团中。图 3.2 中的图有 5 个最大的团：$\{1,2,3\}$，$\{2,3,4,5\}$，$\{3,4,5,6,7\}$，$\{7,8\}$ 和 $\{8,9\}$。下述定理建立了区间图的特征。

定理 3.1[128]　设 G 为一张无向图，以下陈述是等价的。

（1）G 是一个区间图。

（2）G 是一个三角剖分图且它的补图是可比图。

（3）G 的最大团可以线性的排列如下，对 G 中的每个顶点 x，包含 x 的最大团连续出现。

证明　(1)→(2) 遵循引理 3.4 和引理 3.5。

(2)→(3)　设 $G(V, E)$ 是一个三角剖分图，并设 F 为补图 $\bar{G}(V, \bar{E})$ 的可迁定向。又设 A_1 和 A_2 为 G 的最大团。显然，存在 F 中的一条边，一个端点在 A_1 中，另一个在 A_2 中（不然的话 $A_1\cup A_2$ 可以形成 G 中一个团）。很容易看出，连接 A_1 和 A_2 中 $\bar{E}$ 的所有边都有相同的定向。［提示：如果边 (a_1, a_2) 和 (a_1', a_2') 连接 A_1 和 A_2 就如图 3.5（a）所示，那么边 (a_1, a_2') 和 (a_1', a_2) 中至少有一条边属于 $\bar{E}$。取哪个是它的方向呢?］根据 F 中边的方向对最大团排序：$A_1<A_2$，当且仅当存在 F 中的一条边连接 A_1 和 A_2 且朝向 A_2。称这样的定向是可迁的，从而定义了团的线性次序。

假设 $A_1<A_2$ 且 $A_2<A_3$。那么存在 F 中的边 (a_1, a_2') 和 (a_2'', a_3)，其中

$a_1 \in A_1$，a_2'，$a_2'' \in A_2$，$a_3 \in A_3$，如图 3.5（b）所示。如果（a_2'，a_3）$\notin E$ 或者（a_1，a_2''）$\notin E$，那么（a_1，a_3）$\in F$ 且 $A_1 < A_3$。因此，假定边（a_1，a_2''），（a_2''，a_2'）和（a_2'，a_3）都在 E 中。因为 G 不包含无弦的长度为 4 的回路，（a_1，a_3）$\notin E$，并且 F 的可迁性表明（a_1，a_3）$\in F$，因此 $A_1 < A_3$，证明团以线性次序排列。

设最大团以线性次序排列 A_1，…，A_m。假设存在团 $A_i < A_j < A_k$ 其中 $x \in A_i$，$x \notin A_j$ 且 $x \in A_k$。因为 $x \notin A_j$，又顶点 $y \in A_j$，使得（x，y）$\notin E$。但是 $A_i < A_j$ 表明（x，y）$\in F$，而 $A_j < A_k$ 则表明（y，x）$\in F$，于是矛盾。

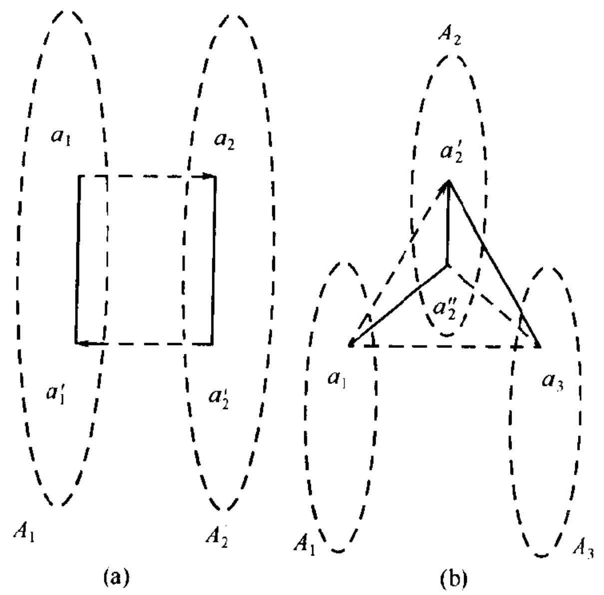


图 3.5 定理 3.1 的证明

(3)→(1) 对每个顶点 x，令 $I(x)$ 表示 G 的所有包含 x 的最大团的集合。对于 $x \in V$，集合 $I(x)$ 构成了区间图 G 的区间。 ■

定理 3.1 将识别区间图的问题归结为识别三角剖分图和可比图的问题[114,227]。

3.5 用限制酶切片段指纹图谱作图

最简单的用限制酶切片段指纹图谱来作图的例子是单完全酶切（single complete digest，SCD）作图[257]。在此情形中，克隆的指纹图谱是单酶切所产生的酶切片段大小的多值集合（multiset）。一个 SCD 图（见图 3.6）是克隆以及与给定的 SCD 数据一致的酶切片段的定位图[127]。

SCD 作图问题 寻找一个与 SCD 数据一致的最简洁的图（即一个具有最少酶切片段数的图）。

寻找最简洁图谱的问题是 NP 难题（NP-hard）。在实践中，克隆的指纹图谱常常提供很强的统计证据，可以用来估计克隆的排序。Jiang 和 Karp[179] 研究了寻找给定排序的克隆的最简洁图谱的问题。

假定没有一个克隆是包含其他克隆的，且假定每个克隆都开始和结束于酶切位点。Jiang 和 Karp[179]将已知克隆排序的 SCD 作图表述为在一个特定的多级图上的约束路径覆盖问题（constrained path cover problem）。令 $\mathcal{S} = \{S_1, \cdots, S_n\}$ 为 SCD 作图的一个实例，其中每个 S_i 是一个多值集合，它代表了从左端点算起的克隆排序中第 i 个克隆的指纹图谱。一个标记的多级图 G（叫做克隆片段

图）由 n 个级组成，其中第 i 级包含 $|S_i|$ 个端点。在级 i，对 S_i 的每个元件 x 都有一个顶点（包括重复的），标记为 x。两个端点是相连的，如果它们处于邻近的级且有相同的标号（见图 3.6）。

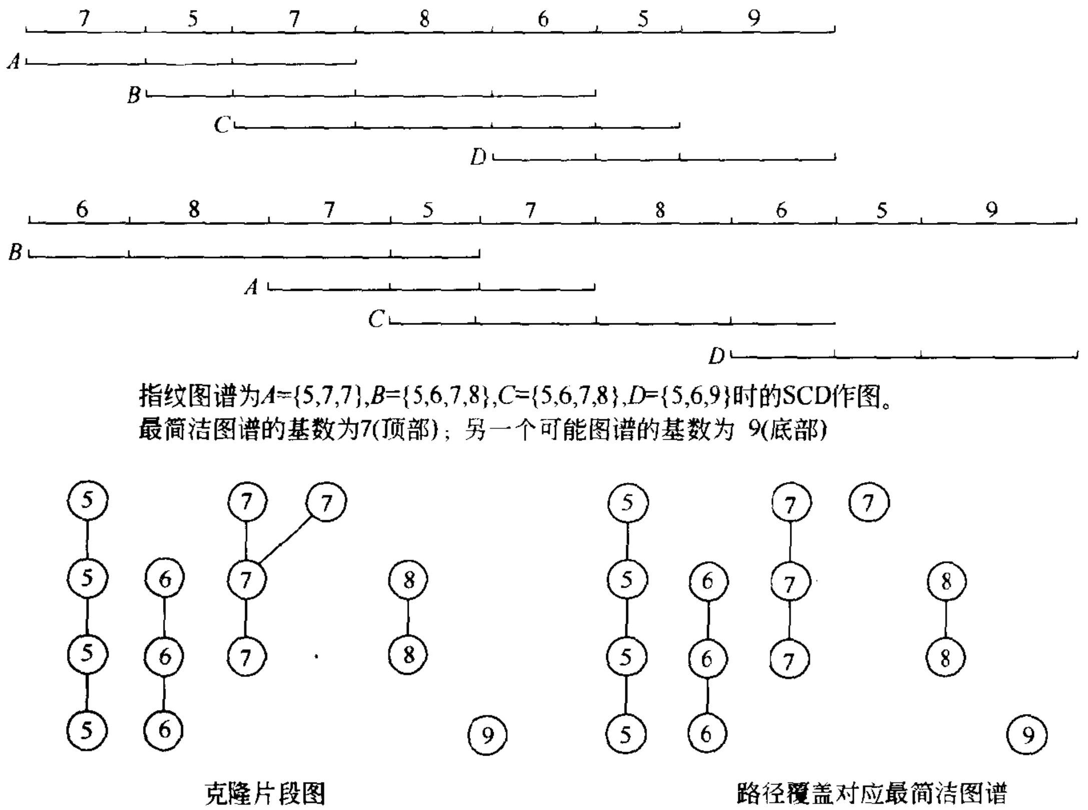


图 3.6 SCD 作图

直观地，G 中的一个路径指定若干连续克隆上的酶切片段的集合，它们可定位于图谱中相同位置。将经历级 i，…，j 的路径简单地表示为 $[i,j]$。

路径覆盖（path cover）是指路径的选择，使每个顶点正确地包含在一条路径中。显然，任何 $\mathcal{S}$ 的图谱相应于有相同基数的 G 的路径覆盖（如图 3.6 所示基数为 7 的路径覆盖）。但是反之并不正确，例如，在图 3.6 中的克隆片段有一个基数为 6 的路径覆盖，但是并不同任何图谱相符。原因在于一些路径是冲突的（与引理 3.1 比较）。如果 $i<i'<j'<j$，路径 $[i,j]$ 和 $[i',j']$ 是冲突的。如果它没有包含冲突的路径，一条路径覆盖是无冲突的。

G 的一条路径覆盖是相容的，如果它相应于有相同基数的 $\mathcal{S}$ 的图谱，类似于引理 3.1。

引理 3.6 当且仅当路径是无冲突的，该条路径覆盖是相容的。

因此，构造一个 $\mathcal{S}$ 的最简洁图谱等价于寻找 G 的最小无冲突路径覆盖。尽管寻找 G 的最小无冲突路径覆盖同最短覆盖字符串问题相似，但是它们计算的

复杂性是很不同的。Jiang 和 Karp[179]描述了一种对给定克隆次序的 SCD 作图的 2-近似算法。

3.6 若干其他的问题和方法

3.6.1 Lander-Waterman 统计

当物理作图计划启动的时候，生物学家们必须决定构造一张图谱需要多少克隆。最少的需要量就是这些克隆可以覆盖几乎全部基因组。在最早期的作图计划中，Olson 等[257]构造了一个包含 $n=4946$ 个克隆的克隆文库（library）。每个克隆携带平均长度为 $L=15000$ 个核苷酸的一条 DNA 片段。这个克隆文库代表了长度为 $G=2\times10^7$ 的 DNA 分子；也就是说每个核苷酸平均由 $\frac{NL}{G}\approx4$ 个克隆来表示。而 $c=\frac{NL}{G}$ 被称为克隆文库的覆盖数。一个典型的文库提供了范围在 5～10 之间的覆盖数。当这项计划开始时，并不很清楚酵母基因组可以被 4946 个克隆覆盖的百分比。

Lander 和 Waterman[214]研究了一种概率模型，其中相同长度的克隆被定位于 DNA 上独立且随机的位置。定义一个缺口为区间，它不被任何克隆所覆盖，而一个重叠群则是没有缺口的最大区间。Lander 和 Waterman[214]证明预缺口的期望数 $\approx ne^{-c}$，而 DNA 不被任何克隆覆盖的期望片段数 $\approx e^{-c}$。Olson 等[257]的作图工作中产生了 1422 个重叠群，同 Lander-Waterman 1457 个重叠群的估计很接近。Arratia 等[11]进一步发展了 Lander-Waterman 的统计方法，并将其应用于杂交指纹图谱的情况。

3.6.2 筛选克隆文库

用原始的方法要获得 n 个克隆和 m 个探针的杂交矩阵（hybridization matrix）需要 $n\times m$ 次杂交实验。构建克隆池（pool）的目的在于减少获得杂交矩阵所需要的实验次数。如果在杂交矩阵中第一个数字很小（如惟一探针作图的情形），那么很多实验会返回负的结果。显然，将克隆池分成一些组并针对这些克隆池测试探针可节省实验开销。

简单地假设 $\sqrt{n}$ 是一个整数，并将 n 个克隆看成 $\sqrt{n}\times\sqrt{n}$ 阵列的元素。相应于这阵列的每行和每列，将克隆组合成克隆池。那么结果的集合由 $2\sqrt{n}$ 个克隆池组成，每个克隆池包含 $\sqrt{n}$ 个克隆。这使杂交实验的次数从 $n\times m$ 次减少到 $2\sqrt{n}\times m$ 次，但是这使得图谱装配的计算问题更加困难[25,98]。Chumakov 等[68]使用这种克隆池策略来构造第一张人类基因组物理图。克隆池策略的计算分析与下

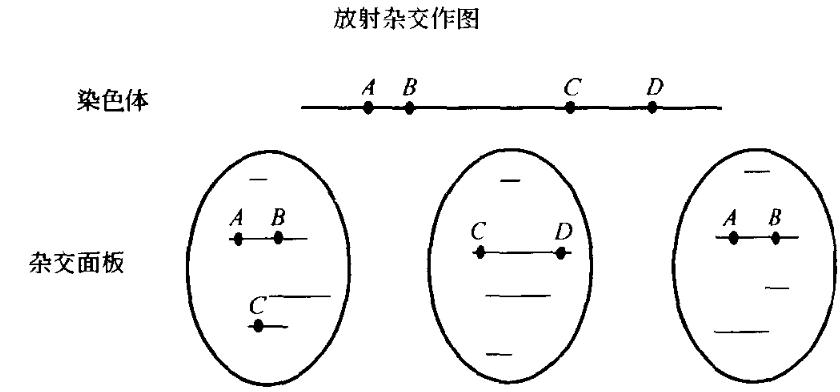
列问题相关。

组测试问题 通过最少次形如“集合 $Q\subset L$ 包含一个特异的对象吗?”的询问，寻找到对象集合 L 中的特异成员。

询问相当于用一个探针来测试克隆池（克隆集合）。在作图的应用中，在相似物中提问是最划算的（不适应组测试）。Bruno 等[51]提倡一种“随机 k-集合设计”的克隆池策略，其优于行-列克隆池设计。在“随机 k-集合设计”中，每个克隆出现在 k 个克隆池中，而且所有 k 个克隆池的选择是等可能的。关于不适应组测试的误差分析可参见文献[200]。

3.6.3 放射杂交作图

放射杂交（RH）作图[77]是一种实验策略，它使用以自然形式出现的随机克隆池。RH 作图包括将人类细胞暴露于放射线，将每条染色体断裂成随机的片段。然后通过融合人类细胞和仓鼠细胞将人的 DNA 片段的随机子集融入仓鼠的染色体来“营救”出这些片段。可以将人的 DNA 片段视为克隆，而将仓鼠细胞看成是由这些克隆构成的克隆池。最后得到的杂交细胞可以长成为包含由人类基因组片段构成的克隆池的细胞株。图 3.7 展示了一个 RH 作图实验，它有 3 个杂交细胞株和 4 个探针（标记），其结果产生一个 4×3 阶的杂交筛选矩阵。放射杂交作图问题是通过杂交筛选矩阵重构标记顺序[317]。



杂交筛选向量				
	A	+	−	+
	B	+	−	+
	C	−	+	−
	D	+	+	−

图 3.7 杂交筛选矩阵

如果放射线引起的切割均匀地在染色体上的随机位置出现，那么在两个邻近标记（像 A 和 B）之间的切割是很少的。这意味着它们是共保留的，也就是说杂交细胞要么两者都保留要么两者都不保留。此观察使我们可说明邻近地定位标

记及使用的理由类似于遗传作图中的探针排列次序[39,215]。Slonim 等[317]对 RH 作图使用了隐马氏模型的方法，并通过检测三个一组的标记建立了一个 RH 图谱的框架。三个一组的标记最可能的次序可以从杂交筛选矩阵估计出来。图 3.7 的例子会产生 *ABC*，*ABD*，*ACD* 和 *BCD* 三个一组的子集合，且问题归结为寻找以这些三联组为子序列的字符串（*ABCD*）。如果出现次序不正确的三联组和不知道方向的三联组［对于三联组 *ABC* 而言，标记次序（*A-B-C* 或 *C-B-A*）的方向是未知的］，那么问题就会变得复杂。

4 测　　序

4.1 引言

在人类基因组计划将要完成的时候，很少有人还记得在 DNA 测序开始之前，科学家们例行公事般地对蛋白质测序。Frederick Sanger 就是因为确定了胰岛素（用于治疗糖尿病的蛋白质）的氨基酸序列而获得了他的第一个诺贝尔奖。在 20 世纪 40 年代后期测序胰岛素比今天人们测序整个细菌的基因更有挑战性。那个时候关于蛋白质测序的计算状况与现在的 DNA 测序的计算状况相类似。主要的不同是测序片段的长度。在 20 世纪 40 年代后期，生物学家开始学习怎样一次从蛋白质尾端截取一个氨基酸，然后读出这个氨基酸。但是，这种方法只对从尾端的几个氨基酸起作用，因为在 4～5 次截断后结果就很难解释了。为了避免这个问题，Sanger 用蛋白酶消化胰岛素，并测序每一个获得的碎片。然后，他用这些交叠的片段来重构整个蛋白质序列，这很像现在的 DNA 测序方法“断裂—读片段—装配”。

Gly	Ile	Val	Glu				
	Ile	Val	Glu	Gln			
				Gln	Cys	Cys	Ala
Gly	Ile	Val	Glu	Gln	Cys	Cys	Ala

这种一次截取和测序一个末端的氨基酸的埃德曼降解技术（Edman degradation）成了此后 20 年的主要的蛋白质测序方法。到了 20 世纪 60 年代后期，市场上出现了蛋白质测序的机器。

Sanger 的蛋白质测序方法影响了 RNA 的测序工作。1965 年，第一个 RNA 测序就是使用了同样的“断裂—读片段—装配”的方法。Holley 和他的合作者在 Cornell 大学用了 7 年的时间来确定 tRNA 中的 77 个核苷酸的序列。此后许多年，DNA 测序是通过先将 DNA 转录为 RNA，然后再测序 RNA 来完成的。

英国剑桥大学和麻省理工学院同时并独立地发明了 DNA 测序的方法。1974 年，前苏联科学家 Andrey Mirzabekov 在访问哈佛大学的 Walter Gilbert 实验室时，发现了一种能够选择性地在 A 和 G 处断裂 DNA 的方法。后来，Maxam 和 Gilbert 发现了在每个 C 和 T 处断裂 DNA 的方法。在测量了 4 个独立反应所获得的片段的长度后，人们能够测定出 DNA 序列了。

Sanger 方法利用了细胞复制 DNA 的方式。细胞一个字母一个字母地复制

DNA，一次增加一个碱基。Sanger 想到可以对不同大小的 DNA 片段制作序列阶梯，如果他“饿死”在生成 DNA 时需要的四种碱基之一的反应。细胞将会复制 DNA 直到用尽某一种碱基。例如，对序列 ACGTAAGCTA，如在 T 处终止反应，则将会生成片段 ACG 和 ACGTAAGC。通过对 A、T、C 和 G 做四种终止实验，并依照长度分离所得到的 DNA 片段，于是可以读出 DNA。而后，Sanger 发现嵌入 A、T、G、C 位置的化学药品能够制止 DNA 链的进一步增长。其结果是在 1977 年发展出了两种独立的 DNA 测序技术[233,297]，并被用于具有 5386个核苷酸的病毒的测序，且在 1980 年获得了诺贝尔奖。从那以后，DNA 测序资料的数量呈指数形增长。1989 年，人类基因组计划启动了。这一计划的目标是确定包含在人类基因组的 30 亿个核苷酸里的大约 100000 个人类基因。在以后的几十年里，遗传文本可能是生物学家的主要研究工具。

与 50 年前的蛋白质测序方法相似，现在的生物学家可以测序较短的（300～500 个核苷酸）DNA 片段，用这些片段来装配连续的基因组。传统的鸟枪测序法起步于长的 DNA 片段（例如长 50kb）的大量复制的样品。样品经过超声处理，把每个片段随机分割为插入片段，而那些太大或太小的片段被删去不予考虑。插入片段的样品被克隆到载体上，随后通过一个细菌宿主感染单个载体。在细菌进行复制后，产生了包含数百万载体的复制品和相关的插入片段的细菌菌落。这样一来，克隆过程导致了待测序的插入片段的样品的产生，这就是 Sanger 等在 1977 年[297]提出的方法。通常，从这个实验里只能解释插入片段的前 300～500 个核苷酸。装配这些片段，就像将许多散乱的活页整理成一本书一样，生物学家必须想法解决这一难题。

4.2 重叠、排列和共有序列

在测序了较短的 DNA 片段（读出片段）之后，要组装这些片段，重构整个克隆的 DNA 序列结构（片段装配问题）。最短的超字符串问题（shortest superstring problem）是一个极度简单的抽象问题，它并不解决实际的片段装配问题，因为它对数据做了完美的假设并忽略了 DNA 的重复现象。人类基因组包含许多重复部分，例如，300bp 的 Alu 序列在人类基因组中重复了（有 5%～15%的变异）大约一百万次。幸运的是，这些重复部分的复制在进化过程中有不同的变异，因此并不是严格的重复。这一发现使研究人员有可能装配序列，甚至对那些有重复部分的序列。

在片段装配问题中的另一个难点是不知道子字符串的方向。DNA 是双链的，两条链中的一条实际上由一个亚克隆来表现，它用任意的方法确定插入片段在载体上的定向。因此在重构的时候是用子字符串还是用它的 Watson-Crick 补体是不知道的。

最早的测序算法伴随着贪婪策略以及把字符串合并在一起（从最重叠的字符串开始）直到只剩下一条字符串为止。许多片段装配的算法包括 3 个步骤[195,262]。

- 重叠　查找潜在的重叠片段。
- 排列　查明片段的顺序。
- 共有　从排列中导出 DNA 序列。

重叠问题是查找一个序列的后缀和另一个序列的前缀的最好的匹配。如果没有测序误差，那么可以很容易地发现一个字符串最长的后缀最好地匹配了另一个字符串的前缀。但是，测序的误差迫使我们在序列比对中使用一种修改的动态规划算法。由于误差很小（1%～3%），通常采取过滤方法，过滤掉那些成对的不共享一个显著地长的子串的片段。

构建排列是 DNA 序列装配中最困难的一步。人们可以把一个片段的 DNA 序列看成这个片段的外延指纹，并利用来自图谱装配的计算思想。许多片段装配算法在每一步选择有最佳重叠的一对片段。重叠的得分或者是相似性得分，或者是在通常的 Phrap 程序中所用的概率得分[136]。对所选的有最佳重叠得分的成对片段做一致性检验，如果这种检验可以接受的话，那么两条片段就可以合并。在算法的后面部分，从重叠连续群（contings）中收集片段来合并而不是单个的片段。排列这一步的困难是确定两条有很好重叠的片段是否就是真的重叠的（即它们的差异是由测序误差造成的）或者代表了一个基因组的重复部分（即它们的差异是由突变产生的）。

构建共有序列的最简单的方法是选择在排列步骤完成后构建的子字符串排列中最频繁出现的字母。更成熟的算法是沿排列在一个小窗口中作子字符串最佳比对。作为一种选择，Phrap[136]通过构造共有序列作为从排列中得到片段的最好的（根据某些概率得分）嵌合物。

4.3 双管鸟枪测序

在 20 世纪 90 年代早期，鸟枪测序方法最早用来测序 50kb 量级的黏性质粒大小的 DNA 片段。结果，人类基因计划最初采用“clone-by-clone”策略，包括物理作图，选取覆盖基因组的最小覆盖克隆集，以及用鸟枪法测序最小覆盖集中的每一个克隆。

在 20 世纪 90 年代中期，DNA 测序方法已推进到可测定整个 1800kb 的流感杆菌（*Haemophilus influenzae*）基因组。受此突破的启发，Weber 和 Myers[367]，提出采用鸟枪法来测序整个人类基因组。1 年后，怀着在 2001 年完成人类基因组的鸟枪法测序的目标，组建了 Celera 基因公司。

但是，在大规模测序计划中，片段组装就变得非常的困难了，例如测序

整个果蝇基因组。这时，标准的片段组装算法注意到定位于基因组不同位置的不稳定重复部分。增加读取的长度将可解决这一问题，但是测序技术仍然没有显著提高读取的长度。为了解决这个问题，生物学家建议了一种有效地增加读取长度的方法，那就是由固定长度缺口的一对检读器来分别读取。在这种方法中，选择大概有相同长度的插入片段，这两个插入片段的两端都会被测序。这就在彼此可知的大致距离内在相对方向产生了一对读出序列（称为配对序列）。

重复片段是全基因组鸟枪法测序的主要挑战。重复片段以不同的规模出现。例如，在人类的T细胞受体座位，存在胰蛋白酶原基因的5个相近定位的重复片段，长度为4kb，并且拷贝间有3%～5%的变异。这些重复部分很难装配，因为在重复部分外部单一检读器无法跨越重复部分。估计人类基因组包含一百万个Alu重复片段（300bp）和200000个长散布核元件（LINE）重复片段（1000 bp），更别提估计有25%的人类基因至少有2个拷贝。

双管DNA测序法的计算优势是插入的两个检读器不太可能都位于大规模DNA重复片段中[286]。因此，在DNA单一部分中，检读器就决定了一个重复片段的拷贝。

通过利用STS图谱，双管鸟枪测序法能更有效地用于片段装配。一个STS是惟一的300bp DNA片段，沿着人类染色体，STS图谱定位了数以千计的STS[173]。因为相邻的STS之间的大概距离是可知的，于是STS的位置可以作为片段装配中的检测点[247,367]。

雄心勃勃的基因组双管鸟枪测序计划对完成鸟枪阶段后如何“结束”剩余的未覆盖部分的测序提出了挑战。任何合理次数的鸟枪法测序都会留下不充分测序的区域。这些区域包括低覆盖的测序区域及不知长度的空缺。通过“步查”(walking)实验来实现结束，也就是使用引物来扩展已测序区域一个读取长度，该引物取自已测序的重叠连续群的。DNA测序的优化要求在鸟枪法测序的次数和步查实验间取得平衡。另外一个问题是，为了满足最小化覆盖准则应在何处做步查实验。

4.4 若干其他的问题和方法

4.4.1 最短超字符串问题

Blum等[37]设计了一个算法，寻找一条不超过最优长度3倍的超字符串。后来，Breslauer等[48]描述了一个算法，它的近似比率为2.596。以最大化重叠合并一对字符串，这一简单的贪婪算法的近似比率问题仍然没有解决。还没有人建立起这样一个例子，可以显示这个算法产生的超字符串超过最优长度的2倍，因

此可以猜测贪婪算法是 2-近似算法。

4.4.2 DNA 测序的结束阶段

产生精确 DNA 序列的最低要求是每一个 DNA 位置至少有 3 个克隆覆盖，并且在共有序列构造处采用多数原则。但是，每一个基因组测序计划都可能导致低克隆覆盖的 DNA 片段。一旦建立了这些区域的位置，在未来的结束阶段人们就要依靠基因组步查实验方法。已测序的 DNA 序列检读的集合定义了一个位置覆盖，那就是在 DNA 序列中覆盖 x 位置的检读次数。假设需要 k 倍的冗余覆盖，结束阶段的目标是找到步查-测序检读的最小集合，它使每一个位置的覆盖增加到 k 倍[78]。

5 DNA 阵 列

5.1 引言

当人类基因组计划（human genome project）启动的时候，DNA 测序还是一项既耗费时间又很难自动化进行的工作。1988 年有 4 个生物学家研究小组同时独立地提出一种全新的测序技术，它被称为杂交测序（sequencing by hybridization，SBH）。SBH 构建一个微型的 DNA 阵列（也称为 DNA 芯片），将数千 DNA 短片段附着在表面上。每个短片段显示一些未知的 DNA 片段的信息，所有的这些信息连接起来就可推断出 DNA 片段的序列。但是在 1988 年几乎没有人相信这个主意能够成功，不但生物化学问题（在阵列上合成数千个 DNA 短片段）而且组合问题（利用阵列输出进行序列重构）看起来都太复杂了。当第一篇介绍 DNA 阵列的文章发表后不久，科学（Science）杂志就合成 DNA 阵列的工作量写道，“可怕的工作将会不断出现”（it would simply be substituting one horrendous task for another）。但这不是一个很好的预测：DNA 阵列技术上的重大突破是由 Fodor 等在 1991 年[110]实现的。他们的 DNA 阵列的制造方法是基于光导多聚体合成法（light-directed polymer synthesis），它同计算机芯片的制造有很多类似的地方。使用这种技术构建一个长度为 l 的 4^l 探针阵列只需要 $4l$ 次单独的反应。使用这种方法，1994 年一家加利福尼亚的生物技术公司 Affymetrix，制作了第一个 64kb 的 DNA 阵列。不久以后制作 1Mb 阵列就成为常规工作了，DNA 阵列的构想从智力游戏变成了一种最有前途的生物新技术，对医学诊断和功能基因组研究产生了巨大的变革作用。

DNA 阵列上的每个探针 p 通过回答 p 是否与靶（未知）DNA 片段杂交来查询靶 DNA 片段。给定一个未知的 DNA 片段，阵列可提供所有包含在这个片段中的所有长度为 l 的字符串的信息（片段的 l 元组成分），但不提供这些字符串的位置信息。然后，组合算法利用片段的 l 元组组成重构片段的序列。

SBH 问题可以看作是 Hamilton 路问题，也就是在图中寻找一条通过每个顶点的路径，并且每个顶点只能通过一次。图的顶点相应于 l 元组，边则相应于重叠的 l 元组对。然而这种归结并没有导致有效的 SBH 算法，因为 Hamilton 路问题的有效算法是未知的。分子生物学研究还没开始的几个世纪之前，SBH 问题已经由——Leonhard Euler——一个伟大的 18 世纪的数学家实际地解决了。当

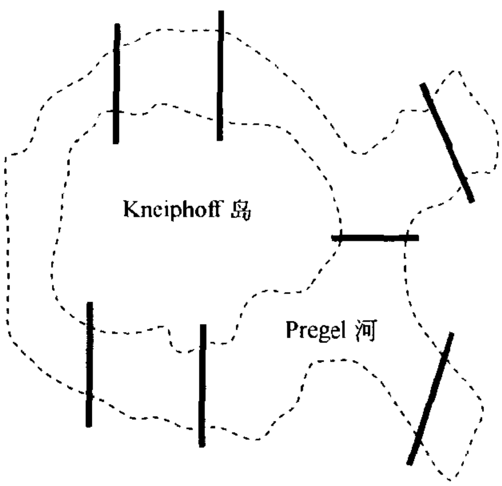


图 5.1 哥尼斯堡桥

然，他并不知道他解决了 SBH 问题，他只是想解决“哥尼斯堡七桥（Seven Bridges of Konigsberg)”难题。哥尼斯堡坐落在由七座桥梁连接的一些岛屿上（见图 5.1），Euler 对找出一条走遍每座桥并且只走一次的路径问题很感兴趣。这个问题的解决预示了图论的诞生，而且在两个世纪之后导致很多组合问题的解决，SBH 也是其中之一。

尽管 DNA 阵列最初提出来是为了取代传统的基于凝胶电泳的 DNA 测序方法，但是用 DNA 阵列从头测序（*de novo* sequencing）仍然是一个难题。DNA 阵列技术应用于 DNA 测序的主要障碍是在杂交数据解释方面的不精确：从非常高的误配中区别出正确的匹配。在从头测序中用短探针（8～10 核苷酸）更是一个特别的难题。

其结果，DNA 阵列在重新测序（re-sequencing）和突变检测（mutation detection）（可以用长探针来做）方面找到比在从头测序方面有更多的应用。在这种情况下，问题就变成了发现（已知的）野生基因和突变（有用的）基因间的差别。长探针（20 核苷酸）可以设计成可靠地检测突变和迂回地解决从非常高的误配中区别出正确匹配的未解问题。这些探针通常是和已知 DNA 片段杂交的探针的变异。例如，DNA 中每 20 元组相应于四种探针：野生型以及 3 个在中心位置上由一个可选核苷酸替换的中间突变型（middle-mutation)。Lipshutz 等[226]设计出一种用来检测 HIV 病毒突变的肽呤阵列(tiling array)。尽管没有区分正确匹配和不正确匹配的杂交信号的阈值，但是通过比较探针的杂交强度和它的中间突变的杂交强度就可以区分这些信号了。

肽呤阵列可以被用来探索整个人类的遗传的多样性。分析在人类线粒体 DNA 上的突变对人类进化和基因疾病的研究有很重大的影响。这些研究包括为寻找突变而在很多个体上进行人类线粒体 DNA 的重新测序。由于传统测序的高昂费用，大多数的研究被限制在大约 600 个碱基对的范围内。Chee 等[65]设计出整个人类线粒体基因组（16569 个碱基对）的肽呤阵列，并且成功地从患有 Leber遗传性视神经病的病人的 mtDNA 样品中检测出 3 个致病的突变。预制 mtDNA阵列可以对许多个体的 DNA 进行重新测序，并且为分子进化的研究[148]提供了快速而有效的技术。DNA 阵列的其他应用包括功能基因组学（检测基因表达）和遗传图[131,349]。

5.2 杂交测序

DNA 阵列或叫 DNA 芯片，几乎同时地由 Bains 和 Smith[23]、Drmanac 等[91]、Lysov 等[228]和 Southern[325]独立地提出。DNA 阵列的发明家们建议将其用来进行 DNA 测序，因此这项技术最初的名字是 DNA 杂交测序（SBH）。SBH 是通过未知的 DNA 片段和大的短探针阵列杂交。给定被称为探针的合成的 DNA 短片段（8～30 个核苷酸），和一条单链靶 DNA 片段，这个靶就会和探针结合（杂交），如果这探针是靶片段的子序列，这个就是 Watson-Crick 的探针互补（A 和 T 互补，G 和 C 互补）。例如，一个探针 ACCGTGGA 会和靶 CCCTG-GCACCTA 杂交，因为靶的子序列 TGGCACCT 和探针是互补的。通过这种方式，探针可以用来检测未知的靶 DNA 并确定它的 l 元组组成。最简单的 DNA 阵列 $C(l)$，包括所有长度为 l 的探针，它是按如下步骤工作（见图 5.2）。

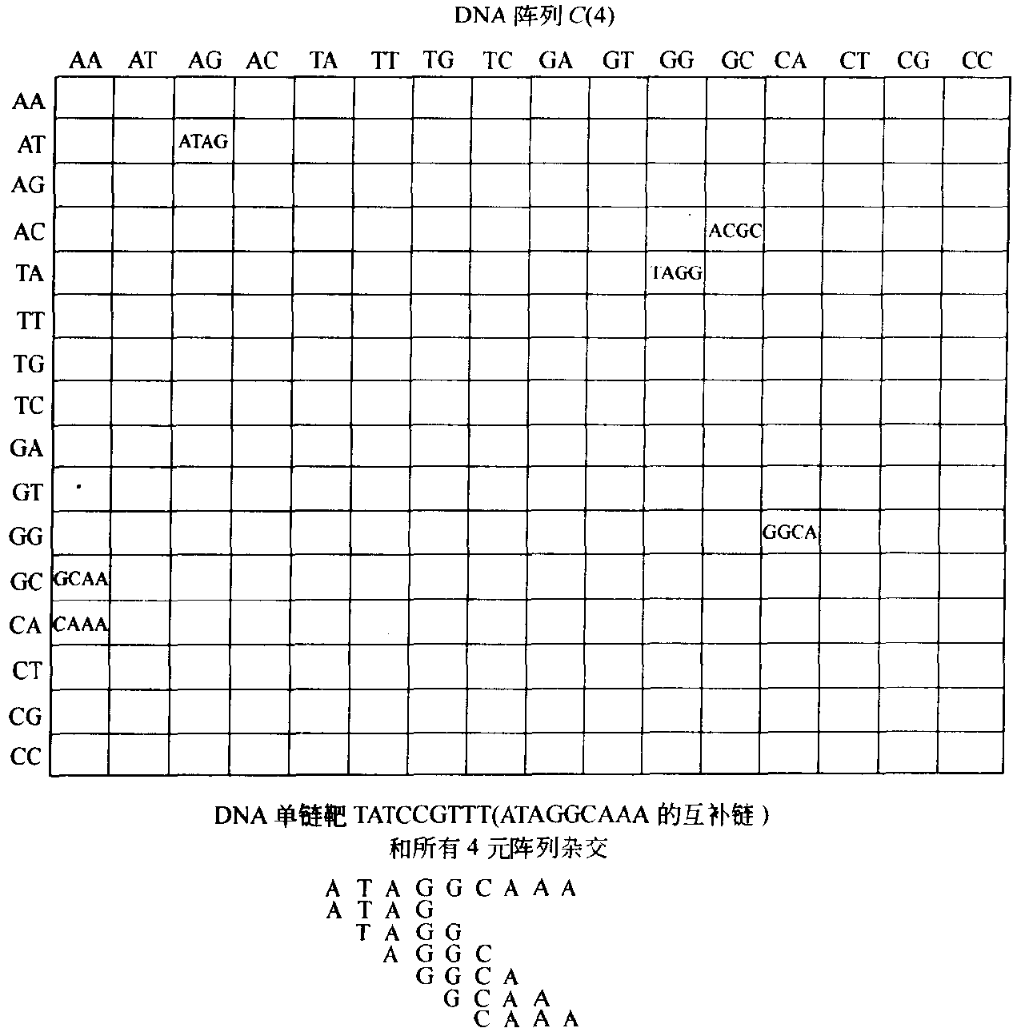


图 5.2 TATCCGTTT 与 DNA 阵列 $C(4)$的杂交

• 将所有可能的长为 l（在第一份关于 SBH 的文章中 $l=8$）的探针附着在

表面，每个探针的位置是确定和已知的，这个探针的集合就称为 DNA 阵列。

• 将包含做过荧光标记的 DNA 片段的溶液涂在阵列上。

• DNA 片段会和那些与片段中长为 l 的子序列互补的探针杂交。

• 检查和 DNA 片段杂交的探针（使用分光检测器）并获得 DNA 片段的 l 元组组成。

• 应用某种组合算法根据 l 元组组成重构 DNA 片段的序列。

“全部 8 元组” DNA 阵列 $C(8)$需要合成 $4^8=65536$ 个探针。在 DNA 阵列首次提出的 1988 年看来是一件可怕的工作。

5.3 SBH 和最短超字符串问题

SBH 提供了 DNA 中的 l 元组的信息，但是并没有提供这些 l 元组的位置信息。假设给定了一个未知字符串（l 元组组成或一个 DNA 片段的谱）的所有长度为 l 的子字符串。那么，怎样根据这些资料来重构靶 DNA 片段？

SBH 可以被看作是最短超字符串问题的特殊情况。对于给定的字符串 $s_1,\cdots,s_m$ 的集合，如果存在一个字符串，使得字符串集合中的每个 s_i 都是它的子字符串，那么这个字符串就称为这个字符串集合的超字符串。给定一个字符串集合，寻找最短超字符串是 NP 完全的（NP-complete）（Gallant 等[117]，1980）。

重叠(s_i,s_j)❶被定义为 s_j 的前缀与 s_i 的后缀匹配的最大长度。最短超字符串问题可归结为在一个完全有向图中的旅行销售员问题，该完全有向图有相应于字符串 s_i 的 m 个顶点和长度为 overlap(s_i,s_j)的边。当所有的子串 $s_1,\cdots,s_m$ 都有相同的长度时，SBH 相应于这种特殊的情形。l 元组 p 和 q 重叠，如果 p 的后 $l-1$ 个字母和 q 的前 $l-1$ 个字母一致，即，overlap$(p,q)=l-1$。给定一个 DNA 片段的谱 S，构造一个有向图 H，有顶点集合 S 和边集合 $E=\{(p,q): p$ 和 q 重叠$\}$。图 H 由前面定义的完全有向图的[长度为$(l-1)$]的边构成。在至少一次访问 H 的每一个顶点的路径和具有谱 S 的 DNA 片段之间有着一一对应关系。

图 5.3 中表现的谱生成路径图 H。在这种情况下，序列 ATGCAGGTCC 重构相应于经过 H 中所有顶点的惟一路径

ATG→TGC→GCA→CAG→AGG→GGT→GTC→TCC

如果只经过图中的每个顶点一次，则这条路径被称为 Hamilton 路。

图 5.4 显示的谱生成一种更复杂的图，它有两条 Hamilton 路，相应于两个可能的重构序列。

对较长的 DNA 片段，重叠图会变得更复杂以至很难分析。Hamilton 路问题是 NP 完全的，目前还没有有效的算法。因此，这种方法对长的 DNA 片段并不

❶ 为方便，仍记为 overlap(s_i,s_j)——译者注

实用。

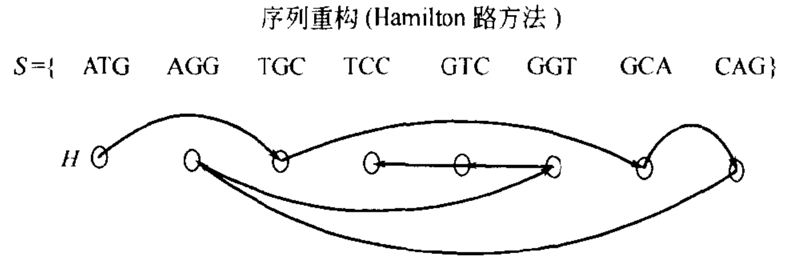


图 5.3 SBH 和 Hamilton 路问题

顶点:谱 S 中的 l 元组。边:重叠 l 元组。

经过每个顶点的路径对应序列重构 ATGCAGGTCC

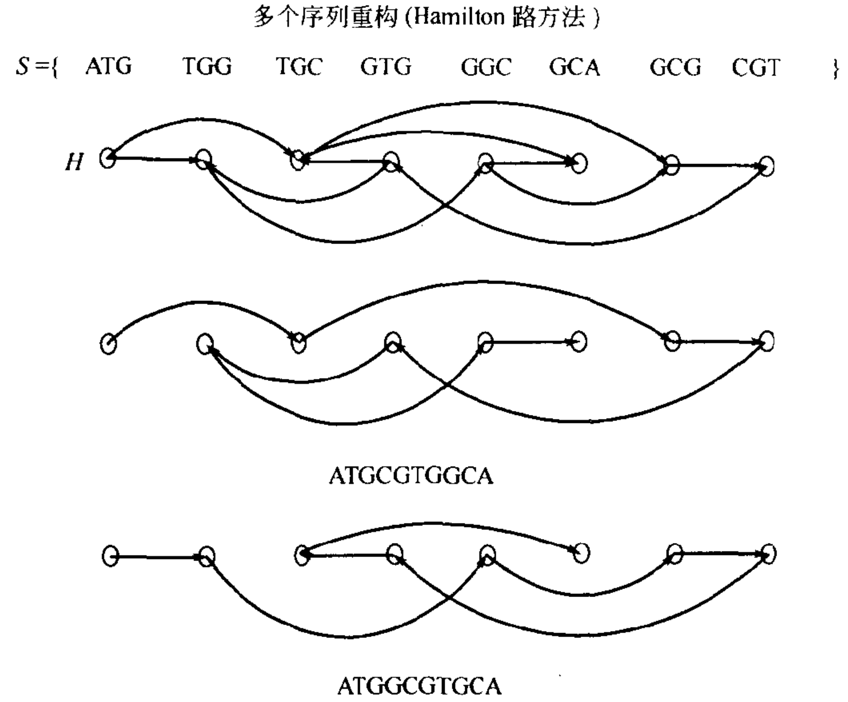


图 5.4 根据两个不同的 Hamilton 路，谱 S 生成两个可能的重构序列

5.4 SBH 和欧拉路问题

正如看到的那样，将 SBH 问题归结为 Hamilton 路问题并没有得到有效的算法。幸运的是，将 SBH 问题归结为欧拉路问题则为序列的重构提供了简单的线性时间算法。该方法的思想是建立一张图，其边是相应于 l 元组（胜于以前构建的顶点），并且找出一条经过每条边一次的路径。在这个方法中，图 G 是依所有 $(l-1)$元组的集合来构建的[264]。一个 $(l-1)$ 元组的 v 由一条方向边与一个 $(l-1)$ 元组的 w 连接起来，如果谱包含一个 l 元组，它的前 $l-1$ 个核苷酸和 v 一致，后 $l-1$ 个核苷酸同 w 一致（见图 5.5）。谱中的每个探针相应 G 中的一条有向边，但不同于 H 中的顶点（比较图 5.3 和图 5.5）。因此，从谱中寻找一条

包含所有探针的 DNA 片段相应于寻找经过 G 的每条边的路径——欧拉路。而寻找欧拉路是众所周知的简单的问题。

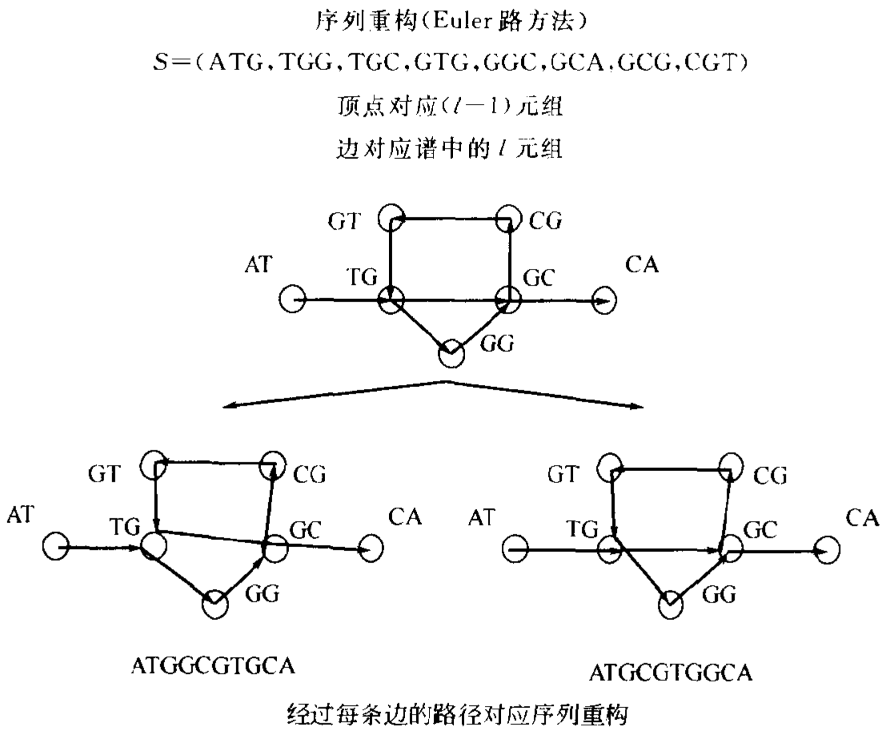


图 5.5 SBH 和欧拉路问题

有向图 G 是欧拉图，如果它包含一个经过 G 中的每条有向边一次的圈。图中的顶点 v 是平衡的，如果进 v 的边数和出 v 的边数一样：$\text{indegree}(v)=\text{outdegree}(v)$[1]。以下定理给出了欧拉图的一些性质。

定理 5.1 一个连通的图是欧拉图，当且仅当它的每个顶点都是平衡的。

SBH 问题等价于在图中寻找一条欧拉路。图中的顶点 v 是半平衡的，如果 $|\text{indegree}(v)-\text{outdegree}(v)|=1$。欧拉路问题可归结为欧拉圈问题，通过在两个半平衡顶点间增加一条边。

定理 5.2 一个连通的图有一条欧拉路，当且仅当它包含至多两个半平衡顶点而其他顶点都是平衡的。

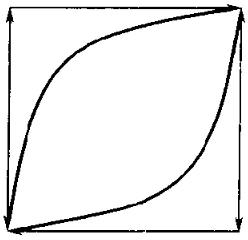
为构建一个欧拉圈，可从 G 中的任意一条边开始，通过与任意的新边连接来扩展已有的迹以形成一条“随机的”迹。当所有与 G 的顶点关联的边都被用于迹中，步骤就结束了。因为 G 中的每个顶点都是平衡的，每条在顶点 v 开始的迹都将在顶点 v 终止。幸运的话这条迹将是欧拉圈，但实际并不需要这样。如果一条迹不是欧拉圈，则它必定包含一个仍然有未遍历边的顶点 w。由于图中所有未遍历边的顶点都是平衡的，因此，存在一条在顶点 w 开始和终止的随机的

[1] v 的入度 $=v$ 的出度——译者注

迹，且它仅包含未遍历的边。现在，可以扩大随机的迹如下：在从 v 开始可到达 w 的随机迹中的一些点插入一条从 w 开始的未遍历边的随机迹。重复这个步骤就可以产生 Euler 圈。这个算法可以在线性时间内实现[108]。

在 SBH 中，不同序列重构的数目由欧拉圈（路）的数目限定。在有向图中有关欧拉圈数目的公式已总结为 BEST 定理[108]。令 G 为邻接矩阵 $A=a_{ij}$ 的有向欧拉图，其中 $a_{ij}=1$，如果有一条边从 G 的顶点 i 到顶点 j，否则，$a_{ij}=0$（见图 5.6）。定义矩阵 M，它是由 $-A$ 的第 i 个对角元替换为入度indegree(i)而得到的，其替换是对所有 i 的。矩阵 M 的 i 余子式是由从 M 中划去第 i 行和第 j 列得到的矩阵 M_i 的行列式 $det(M_i)$决定。矩阵 M 的所有余子式都有相同的值，记为 $c(G)$。

G



$$A=\begin{vmatrix}0&1&0&0\\0&0&1&1\\1&1&0&0\\0&0&1&0\end{vmatrix}\qquad M=\begin{vmatrix}1&-1&0&0\\0&2&-1&-1\\-1&-1&2&0\\0&0&-1&1\end{vmatrix}$$

图 5.6 矩阵 M 的每个余子式是 2。图 G 中欧拉圈的数目是 $2\cdot 0!\ \cdot 1!\ \cdot 1!\ \cdot 0!\ =2$

定理 5.3 在欧拉图 $G(V,E)$中欧拉圈的个数为

$$c(G)\prod_{v\in V}[\text{degree}(v)-1]!$$

对于给定的谱，可用一个简单的方法来检查是否存在惟一的序列重构。将 Euler 图 G 分解为简单的圈 $C_1,\cdots,C_t$，即没有自我交叉的圈。G 中的每条边只能被一个圈使用（G 中的顶点可以被多个圈使用）。从这些圈中定义有 t 个顶点 $C_1,\cdots,C_t$ 的交叉图 C_I，其中 C_i 和 C_j 由 k 条边连接，当且仅当它们有 k 个公共顶点。

定理 5.4 图 G 仅有一个欧拉圈，当且仅当交叉图 G_I 是树。

欧拉路方法工作于无误差的 SBH 实验情况。即使在这种情况下，也会存在多条欧拉路，并导致多个序列重构。在利用 DNA 阵列的实际实验中，谱中的误差使重构更困难。另外，l 长度的重复序列使得分析更复杂，因为从杂交强度很难确定 l 元组的多样性。

图 5.7 给出了和图 5.5 同样的谱，有两个三核苷酸缺失（假阴性）和两个可能的重构序列（其中只有一个是正确的）。当谱包含的 l 元组在靶 DNA 片段中缺失时（假阳性），这种情形在非特异性杂交中变得更复杂。若干种生物化学方法可排除 SBH 实验中的非特异性杂交，以便更好地区分正确和非正确匹配。但是，

DNA 阵列的杂交数据还很模糊，远没达到计算机科学家和生物学家希望的那样。

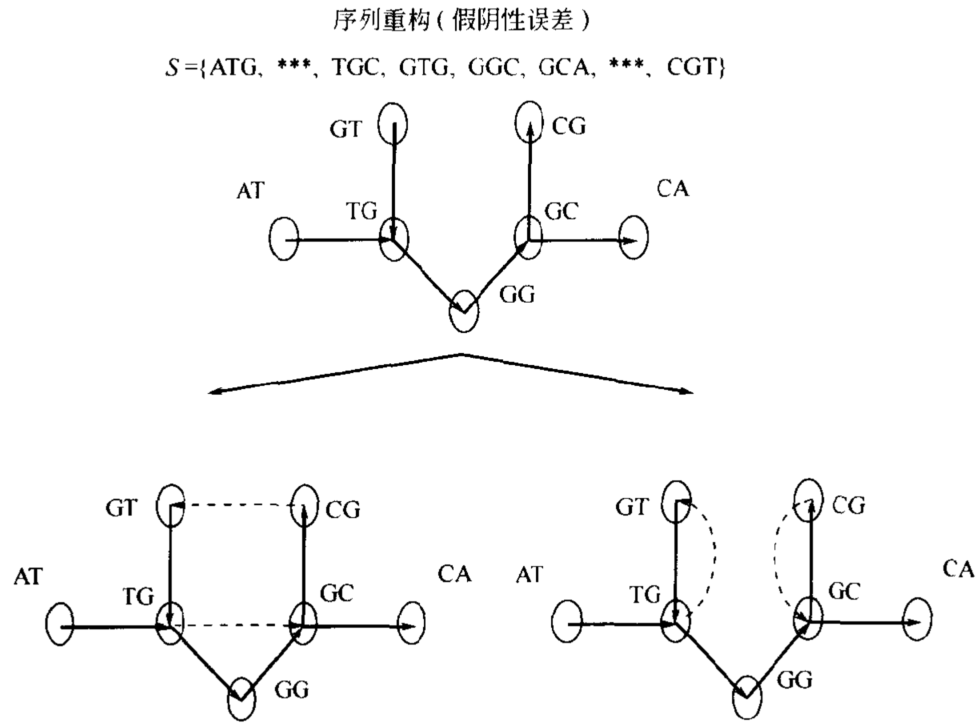


图 5.7 在缺失 l 元组的情况下的序列重构（假阴性）

5.5 惟一序列重构的概率

什么是长度为 n 的 DNA 片段能由一个 DNA 阵列 $C(l)$ 惟一重构的概率？或者换句话说，多大的 l 才可以由它的 l 元组谱惟一重构长度为 n 的随机序列？为简单计，假设 DNA 片段的每个字符都是独立且对每个字符 A、T、G 和 C 取等概率 $p=\frac{1}{4}$。

原始的启发是基于 l 元组重复常导致非惟一的重构的观察资料。相应于长度为 n 的 DNA 片段的一对位置，出现长度为 l 的重复片段的概率约为 $\binom{n}{2}p^l$。为了求得最小的探针长度，使 n 个字符的序列能由它的 l 元组谱可靠地重构，求解 $\binom{n}{2}p^l=1$，产生一个粗略的估计 $l=\log_{\frac{1}{p}}\binom{n}{2}$。可以用 $C(l)$ 阵列重构的 DNA 片段的最大长度粗略的估计为 $\sqrt{2\times 4^l}$。

更仔细的分析表明 l 元组的重复并不是导致非惟一重构的主要原因（Dyer 等[94]，1994）。最主要的原因似是（$l-1$）元组的交叉重复（AGTC 和 TTGG 重复在图 5.9 中的交叉）。因此，长度为 $l-1$ 的重复应予重视。观察显示重复可能形成簇，因此，有些像长度为 $l-1$ 的“最大重复”也应予重视。重复的期望

数大概为 $\lambda\approx\binom{n}{2}(1-p)p^{l-1}$。Poisson 估计取为 $P\{k\text{ 次重复}\}=e^{-\lambda}\frac{\lambda^k}{k!}$。Arratia 等[12]指出，当存在 k 次重复时，没有交叉时的概率为$\approx\frac{k!\ 2^kC_k}{(2k)!}$，其中 $C_k=\frac{1}{k+1}\binom{2k}{k}$ 是第 k 个 Catalan 数。在 k 上求平均值显示，对 n 个字符的序列由其 l 元组谱惟一重构的概率约为

$$\sum_{k\geqslant 0}e^{-\lambda}\frac{k!2^kC_k}{(2k)!}\times\frac{\lambda^k}{k!}=e^{-\lambda}\sum_{k\geqslant 0}\frac{(2\lambda)^k}{k!(k+1)!}$$

Arratia 等[12]将这些启发式论证转变为解决 DNA 阵列的功效的精确估计。

5.6 字符串重排

图 5.8 展示了两个有相同 SBH 谱的 DNA 序列。相应于图 5.8 的谱的图 G 有一个分枝顶点 TG。不知道在最初的序列中到底是哪个 3 元组（TGC 或 TGG）跟在 ATG 的后面，而且很难区分正确和不正确的重构。一个附加的生物化学实验（例如，将靶 DNA 片段同 ATGC 杂交）可能找出正确的重构（图 5.8 左边的序列包含 ATGC，而右边的序列则没有）。

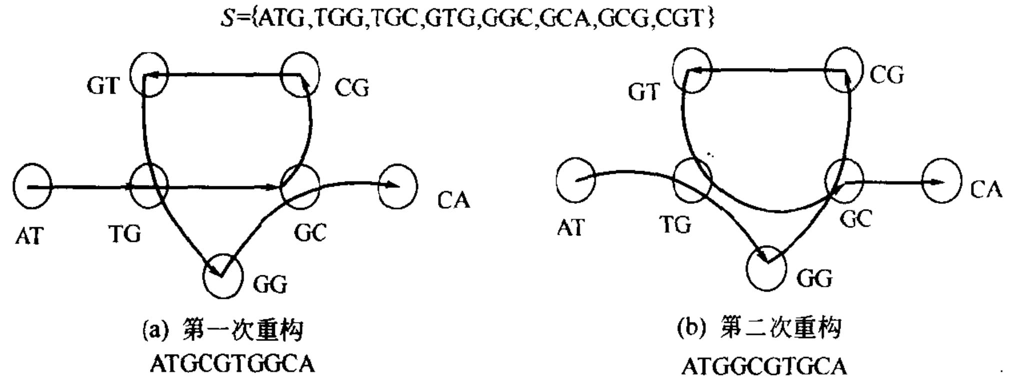


图 5.8 附加的生物化学实验解决了序列重构的不确定性
用 ATGC 做的附加实验找出正确的重构
如果 ATGC 与靶 ATGCGTGGCA 杂交
如果 ATGC 不与靶 ATGGCGTGCA 杂交

在重构过程中解决分枝问题的附加生物化学实验是由 Southern[325]（对每个分枝顶点使用较长的探针）和 Khrapko 等[197]（持续堆积杂交，continuous stacking hybridization）提出的。持续堆积杂交建议用短探针来进行附加杂交，通过靶 DNA 片段和阵列上的探针持续的杂交来扩充双方的序列。在这种方法中，用阵列 $C(l)$上的 m 元组来杂交会提供靶 DNA 序列中的一些（$l+m$）元组的信息。计算机模拟显示只用 3 个附加的实验来持续堆积杂交，97%的情况下会提供 1000bp片段的明确的重构。

分析附加的生物化学实验，需要与给定谱有关的所有DNA序列的表示特征。在DNA阵列研究开始的时候，生物学家认为字符串重排时不会改变字符串的谱[91]。然而，所有重排的表示特征问题仍然没解决。Ukkonen[340]推测有相同l元组谱的两条字符串可以相互转换，只要通过很简单的叫做对换和旋转的变换（见图5.9)。以后，"一条字符串写成l元组记法"，其意思是字符串由它的l元组序列表示。例如，字符串ATGGGC的2元组记法是AT TG GG GG GC。Ukkonen[340]推测任何两个由相同l元组组成的字符串通过叫做对换和旋转的简单操作可以相互转换。

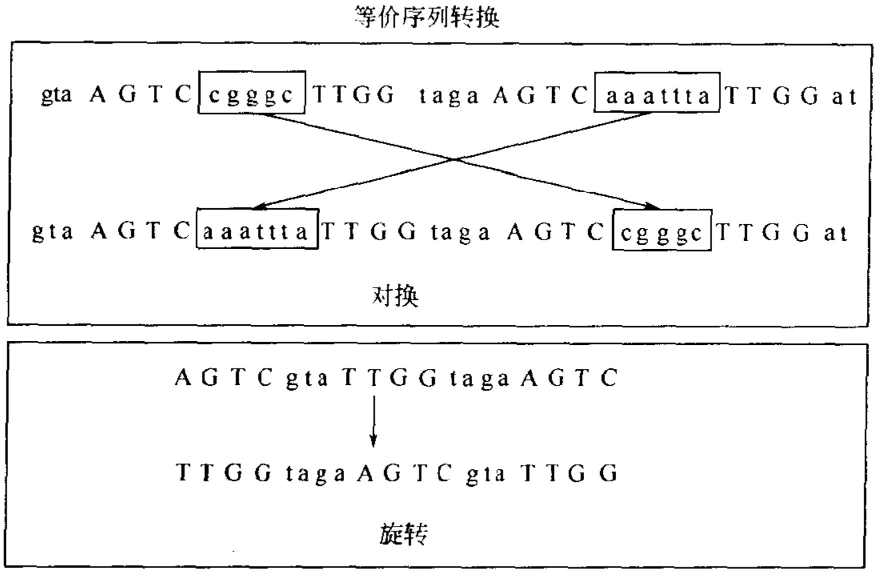


图5.9 Ukkonen变换（$l=5$）

如果一个字符串

$$s=\cdots x\,\underline{\cdots}\,z\cdots x\,\underbrace{\cdots}\,z\cdots$$

[写成$(l-1)$元组记法] 包含$(l-1)$元组x和z的交叉对，那么，字符串$\cdots x\,\underbrace{\cdots}\,z\cdots x\,\underline{\cdots}\,z\cdots$，其中$\underline{\cdots}$和$\underbrace{\cdots}$交换位置被称为$s$的一个对换。如果$s=\cdots x\,\underline{\cdots}\,x\,\underbrace{\cdots}\,x\cdots$，其中$x$是一个$(l-1)$元组。也称$\cdots x\,\underbrace{\cdots}\,x\,\underline{\cdots}\,x\cdots$是一个$s$的对换。如果字符串

$$s=x\,\underline{\cdots}\,z\,\underbrace{\cdots}\,x$$

[写成$(l-1)$元组记法] 开始和结束都是同一个$(l-1)$元组x。则字符串$s=z\,\underbrace{\cdots}\,x\,\underline{\cdots}\,z$被称为是$s$的一个旋转。

显然，对换和旋转并不改变l元组组成。

定理5.5[267] 两个有相同l元组组成的字符串可以通过对换和旋转相互转换。

证明 给定l元组组成的字符串相应于有向图G中的欧拉路（见图5.10)。

图 G 或者是欧拉图或者包含有欧拉路。注意到仅当 G 是个欧拉图时才存在一个相应字符串的旋转。在这种情况下，旋转简单地相应于一个欧拉圈的初始顶点的选取。

G 中的每条有向边 $a=(v,w)$ 用两条（无向）边替换，(v,a) 涂白色 (a,w) 涂黑色（见图 5.10）。显然，新图 G^* 中每条交错的路都是 G 中的有向路，反之亦然。根据定理 2.2，顺序的改变和反射产生 G^* 中的所有 Euler 路，于是生成所有给定 l 元组组成的字符串。注意到 Ukkonen 的对换相应于 G^* 中的顺序的交换。换句话说，G^* 中的每个圈是偶的，因此在 G^* 中就没有顺序反射。这就证明了 Ukkonen 的推测，通过对换和旋转可生成具有给定 l 元组组成的所有字符串。

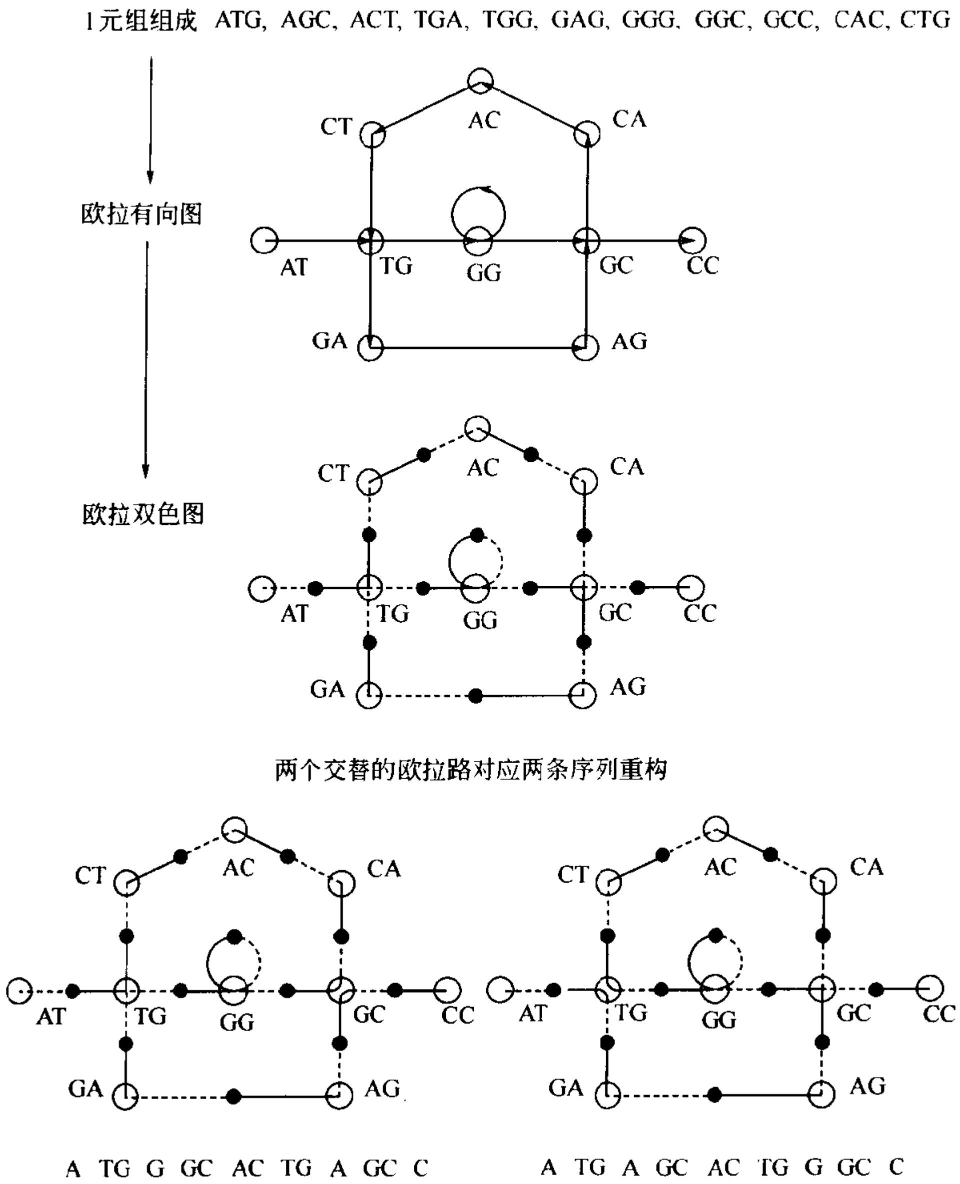


图 5.10 Ukkonen 猜想

右边的序列（欧拉路）是由左边的序列通过对换而得到的，此对换由两核苷酸 TG 和 GC 的交错对定义。

5.7　2-最优欧拉圈

有向图中的 2-路(v_1, v_2, v_3)是指由两条有向边(v_1, v_2)和(v_2, v_3)构成的路。图 $G(V,E)$中每条欧拉圈定义与圈中每对连续边相应的$|E|$条 2-路的集合。$|E|$条 2-路的集合是有效的，如果 E 中的每条边在这个集合中出现两次：一次在 2-路的开头，一次在 2-路的末端。每个有效的 2-路集合定义一个分解，它将 E 分解为圈。定义一个欧拉圈的有效的 2-路集合称为 2-路欧拉集。

设 G 是一个有向欧拉图，它将权重 $w(v_1v_2v_3)$指定给 G 中的每个 2-路($v_1v_2v_3$)。欧拉圈 $C=v_1\cdots v_m$ 的权重等于它的 2-路的权重之和 $w(C)=\sum_{i=1}^{m-1}w(v_iv_{i+1}v_{i+2})$（假设 $v_m=v_1, v_{m+1}=v_2$）这个 2-最优欧拉圈问题就是寻找最大权重的欧拉圈[147]。

设

$$C=\cdots\bar{v}x\,\underline{\bar{w}\cdots\bar{\bar{v}}}\,y\,\bar{\bar{w}}\cdots\hat{v}\,x\,\underbrace{\hat{w}\cdots\hat{\hat{v}}}\,y\,\hat{\hat{w}}\cdots$$

是 G 中交替通过顶点 x 和顶点 y 的欧拉圈。一个顺序交换将 C 变为

$$C'=\cdots\hat{v}\,x\,\underbrace{\hat{w}\cdots\hat{\hat{v}}}\,y\,\bar{\bar{w}}\cdots\hat{v}\,x\,\underline{\bar{w}\cdots\bar{\bar{v}}}\,y\,\hat{\hat{w}}\cdots$$

称 C 在 x,y 的欧拉转换。有向图中的每两个欧拉圈能通过一系列欧拉转换而相互转化[267]。

记 C 在 x,y 的欧拉转换为 $s=s(x,y)$，且记 $C'=sC$。在 x,y，C 的欧拉转换 s 将 C 中的 4 条 2-路

$$\bar{v}x\bar{w},\ \bar{\bar{v}}\,y\,\bar{\bar{w}},\ \hat{v}\,x\,\hat{w},\ \hat{\hat{v}}\,y\,\hat{\hat{w}}$$

变换为 C'中的 4 条 2-路

$$\bar{v}x\hat{w},\ \hat{\hat{v}}\,y\,\bar{\bar{w}},\ \hat{v}\,x\,\bar{w},\ \bar{\bar{v}}\,y\,\hat{\hat{w}}$$

记

$$\Delta_x(s)=w(\bar{v}x\hat{w})+w(\hat{v}x\bar{w})-w(\bar{v}x\bar{w})-w(\hat{v}x\hat{w})$$

$$\Delta_y(s)=w(\hat{\hat{v}}y\bar{\bar{w}})+w(\bar{\bar{v}}y\hat{\hat{w}})-w(\bar{\bar{v}}y\bar{\bar{w}})-w(\hat{\hat{v}}y\hat{\hat{w}})$$

且 $\Delta(s)=\Delta_x(s)+\Delta_y(s)$，其中 $\Delta(s)$ 为欧拉圈 C 和 C'的权重之差。如果 $w(sC)>w(C)$，那么欧拉转换 s 对圈 C 是渐增的。

考虑一个 2-路的有效集合 S，它包含四条 2-路

$$\bar{v}x\bar{w},\ \bar{\bar{v}}\,y\,\bar{\bar{w}},\ \hat{v}\,x\,\hat{w},\ \hat{\hat{v}}\,y\,\hat{\hat{w}}$$

这个集合在顶点 x,y 的转换是一个新的 2-路有效集合，它用下列 2-路集替换上面的四条 2-路

$$\overline{v}x\hat{w},\hat{\hat{v}}y\overline{\overline{w}},\hat{v}x\overline{w},\overline{\overline{v}}y\hat{\hat{w}}$$

如果 S 是一个 2-路的欧拉集合，那么，S 的一个转换被称为非欧拉的，如果它将 S 变换为非欧拉的 2-路集合。例如，根据与欧拉圈 $\cdots\overline{v}x\,\underline{\overline{w}\cdots\hat{v}}\,x\,\hat{w}\cdots\overline{\overline{v}}\,y\,\underbrace{\overline{\overline{w}}\cdots\hat{\hat{v}}}\,y\,\hat{\hat{w}}\cdots$ 相应的 2-路集合在 x、y 的转换是非欧拉转换。

Gusfield 等[147]在 2 进 2 出（2-in-2-out）有向图（即具有最大度为 2 的图）中研究了 2-最优欧拉圈问题。

定理 5.6 在 2 进 2 出图中任意两个欧拉圈能够被相互转化，只要欧拉转换序列 s_1，…，s_t，满足 $\Delta(s_1)\geqslant\Delta(s_2)\geqslant\cdots\geqslant\Delta(s_t)$。

证明 设 s_1，…，s_t 为将 C 变换为 C^* 的欧拉转换序列，且满足向量 $(\Delta(s_1)$，$\Delta(s_2)$，…，$\Delta(s_t))$ 是将 C 变换为 C^* 的所有欧拉转换序列中字典最大的。如果向量不满足条件 $\Delta(s_1)\geqslant\Delta(s_2)\geqslant\cdots\geqslant\Delta(s_t)$，那么 $\Delta(s_i)<\Delta(s_{i+1})$，$1\leqslant i\leqslant t-1$。设 $C'=s_{i-1}\cdots s_1C$。如果转换 s_{i+1} 是 C' 中的欧拉转换（也就是说，由 S_{i+1} 转换产生的 C' 中的 2-路系统的改变定义一个欧拉圈），那么 s_1，…，s_{i-1}，s_{i+1} 比 s_1，…，s_{i-1}，s_i 字典大，于是自相矛盾。（Pevzner[267]指出存在一个将 C 变为 C^* 的具有前缀 s_1，…，s_{i-1}，s_{i+1} 的变换。）因此，在 x,y 的转换 s_{i+1} 不是 C' 的欧拉转换。就是说在 C' 中 x,y 不是交替出现的：$C'=\cdots x\cdots x\cdots y\cdots y\cdots$。另一方面，因为在 C' 中转换 s_i 在 z,u 是一个欧拉转换，所以在 C' 中出现的 z,u 是交替的：$C'=\cdots z\cdots u\cdots z\cdots u\cdots$。需要考虑在 C' 中四个顶点 x,y,z 和 u 所有的交替排列。这情况就是 s_{i+1} 在 $s_i\cdot C'$ 中是欧拉转换的条件（即 s_i 以这样的方式移动 x 和 y 使它们在 $s_i\cdot C'$ 中成为交错的）使得大部分情况都无效（事实上，这是证明的关键想法）。很容易看出，在 C' 中 x,y,z 和 u 所有有效的排列“等价”于排列

$$C'=\cdots z\,\underline{\cdots x\cdots}\,u\cdots x\cdots y\cdots z\,\underbrace{\cdots}\,u\cdots y\cdots$$

在这种情况下，s_i 在两个 y 之间插入 x 使得 x 和 y 交错

$$s_i\cdot C'=\cdots z\,\underbrace{\cdots}\,u\cdots x\cdots y\cdots z\,\underline{\cdots x\cdots}\,u\cdots y\cdots$$

注意到在这种情况下，在 z,y 的转换 s' 和在 x,u 的转换 s'' 都是在 C' 中的欧拉转换。而且 $\Delta(s_i)+\Delta(s_{i+1})=\Delta(s')+\Delta(s'')$。不失一般性，假定 $\Delta(s')\geqslant\Delta(s'')$。于是

$$\Delta(s')\geqslant\frac{\Delta(s')+\Delta(s'')}{2}=\frac{\Delta(s_i)+\Delta(s_{i+1})}{2}>\Delta(s_i)$$

因此，向量 $[\Delta(s_1),\cdots,\Delta(s_{i-1}),\Delta(s')]$ 是字典大于向量 $[\Delta(s_1),\cdots,\Delta(s_{i-1}),\Delta(s_i)]$ 的，是自相矛盾的。 ■

定理 5.6 蕴涵下述定理。

定理 5.7[147] 如果 C 是个非 2-最优的欧拉圈，那么存在一个渐增的 C 的欧

拉转换。

定理 5.6 的证明也蕴涵着当 2-路的所有权重都不相同时，在每一步选择具有最大权重的转换的贪婪算法可得到 2-最优欧拉圈。

5.8 杂交法定位测序

虽然 DNA 阵列最初是为 DNA 测序而提出来的，但是 DNA 阵列解决问题的能力是相当低的。用 64kb 阵列，在单个 SBH 实验中可以重构的 DNA 片段的长度最多是 200bp。为了改进 SBH 解决问题的能力，Broude 等[49]提出了定位 SBH（PSBH），提供（利用附加的实验）在靶 DNA 片段中每个 l 元组的近似位置的测量。尽管这使重构更加明确，但是 PSBH 序列重构的多项式算法还不知道。PSBH 可归结为计算附加约束的欧拉路，如所计算的欧拉路中的边的位置应当在与边相联系的位置区域中。

PSBH 推动了定位欧拉路问题的研究。输入到定位欧拉路问题的是欧拉图 $G(V, E)$，其每条边都与一个整数区域相关联，问题是在 G 中寻找一条欧拉路 e_1，…，$e_{|E|}$，使得 e_i 的区域中包括 i。

定位欧拉路问题　给定有向重图 $G(V, E)$ 和与每条边 $e\in E$ 相关联的区间 $I_e=\{l_e, h_e\}$，$l_e\leqslant h_e$，在 G 中寻找一条欧拉路 e_1，…，$e_{|E|}$，使得 $l_{e_i}\leqslant i\leqslant h_{e_i}$，$1\leqslant i\leqslant |E|$。

Hannenhalli 等[156]已说明定位欧拉路问题是 NP 完全的。在一些明确的记录中，使用多项式算法去解决特殊情况的 PSBH，其中任何边的允许位置区域被常数限定（在 PSBH 中位置的准确实验测量）。

Steven Skiena 提出了 PSBH 问题的稍微有所不同的设计，其模拟实验数据更适合。根据这种新的设计，前面章节叙述的 2-最优欧拉路算法提供了一个解决方案。

实验 PSBH 数据提供了 l 元组近似位置的信息，但是通常没提供误差范围的信息。其结果是，取代与每条边都关联的区间 $\{l_e, h_e\}$ 的，仅是与每条边相关联的近似位置 m_e。在 PSBH 问题的一个不同但更适当的设计中，其目标是使 $\sum_{i=0}^{E}\left|(m_{e_{i+1}}-m_{e_i})\right|$ 最小，其中 $m_{e_0}=0$，$m_{e_{|E|+1}}=|E|+1$。对 G 中每对连续的边 e，e'，定义与 2-路相应的权重为 $|m_e-m_{e'}|$。于是，PSBH 问题就是寻找一条权重最小的 2-最优欧拉路。

5.9 DNA 阵列的设计

因为 DNA 阵列上的特征数是固定的，所以对设计探针的最小集合感兴趣，它足够应付给定长度的几乎所有序列的测序。假定一个 DNA 阵列上的

位置 m 数是固定的，而问题是设计 m 个探针使 DNA 阵列能够提供最大的分辨率。生产包含所有的 l 元组的均匀阵列 $C(l)$ 是多余的。Pevzner 和 Lipshutz[271]介绍了与均匀阵列相比改进了分辨率的新阵列。这些阵列是基于将探针集合成多探针（multiprobe）的想法：在阵列的每个地址上合成一组不同的探针。多探针是位于阵列的专一地址上的一组探针。DNA 片段与多探针杂交，如果 DNA 片段至少和多探针中的一个探针杂交。例如，WWS 是一个包含 8 个探针的多探针

AAG，AAC，ATG，ATC，TAG，TAC，TTA，TTC

（W 代表 A 或 T，S 代表 G 或 C）。RYR 是一个包含 8 个探针的多探针

ATA，ATG，ACA，ACG，GTA，GTC，GCA，GCG

（R 代表嘌呤 A 或 G，Y 代表嘧啶 G 或 C）。TXG 是一个包含四个探针的多探针

TAG，TTG，TGG，TCG

（X 代表任何核苷酸 A，T，G 或 C）。

现在，一个阵列被定义为多探针的集合 C，每个多探针是一组探针。阵列的内存 $|C|$ 是 C 中多探针的数目。每条 DNA 序列 F 定义一个由多探针集与 F 杂交而组成的阵列 C 的子集（C 中 F 的谱）

$$F_C=\{p\in C\text{:多探针 } p \text{ 包含在序列 } \bar{F} \text{ 中出现的探针}\}$$

（$\bar{F}$ 表示 F 的互补序列）。

二元阵列 $C_{\text{bin}}(l)$是内存为$|C_{\text{bin}}(l)|=2\cdot 2^l\cdot 4$ 的阵列，它由两类多探针的全体构成

$$\underbrace{\{W,S\},\{W,S\},\cdots,\{W,S\}}_{l},\{N\}\text{和}\underbrace{\{R,Y\},\{R,Y\},\cdots,\{R,Y\}}_{l},\{N\}$$

其中 N 为指定的核苷酸 A、T、G 或 C。每个探针是由 2^l 个长度为 $l+1$ 的探针的混合物。例如，阵列 $C_{\text{bin}}(l)$ 是由 16 个多探针组成

WA，WC，WG，WT，SA，SC，SG，ST，RA，RC，RG，RT，YA，YC，YG，YT

每个多探针是由两个二核苷酸组成（见图 5.11）。

缺口阵列 $C_{\text{gap}}(l)$[272]是内存为$|C_{\text{gap}}(l)|=2\cdot 4^l$，由两种类型的所有多探针组成

$$N_1N_2\cdots N_l \text{ 和 } N_1N_2\cdots N_{l-1}\underbrace{XX\cdots X}_{l-1}N_l$$

其中 N_i 为指定的碱基，而 X 则为非指定的碱基。每个第一种类型的多探针仅由长度为 l 的一个探针组成；每个第二种类型的多探针是由长度为 $2l-1$ 的 4^{l-1} 个探针组成。

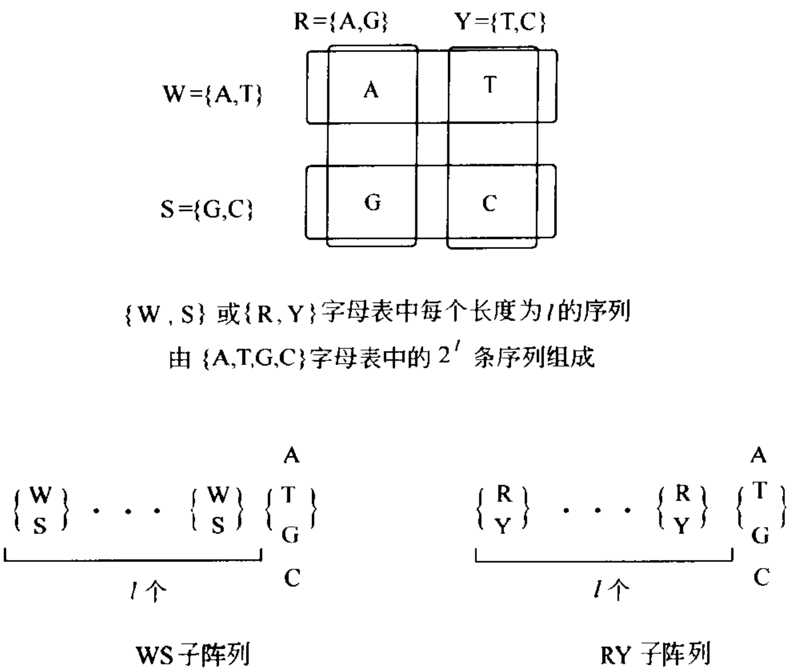


图 5.11 二元阵列

交替阵列 $C_{alt}(l)$ 是内存为 $|C_{alt}(l)|=2\cdot 4^l$ 的阵列，它由两种类型的所有多探针组成

$$N_1XN_2X\cdots N_{l-2}XN_{l-1}XN_l \text{ 和 } N_1XN_2X\cdots N_{l-2}XN_{l-1}N_l$$

每个第一种类型的多探针是由长度为 $2k-1$ 的 4^{k-1} 个探针组成，每个第二种类型的多探针是由长度为 $2k-2$ 的 4^{k-2} 个探针组成。

5.10 DNA 阵列的分辨率

考虑序列 $F=X_1\cdots X_{m-1}X_mX_{n+1}\cdots X_n$，且假设它的前缀 $F_m=X_1X_2\cdots X_m$ 已经被重构了。估计明确地将前缀向右延伸一个核苷酸的概率。

因为 F_m 是 F 的前 m 个核苷酸的可能的重构

$$(F_m)_C\subset F_C$$

有四种方法延伸 F_m：F_mA、F_mT、F_mG 和 F_mC。定义由核苷酸 N 得到的 F_m 的延伸为一个可能的延伸，如果

$$(F_m\mathrm{N})_C\subset F_C \tag{5.1}$$

称序列 F 关于阵列 C 在 m 个核苷酸后是可延伸的，如果条件式 (5.1) 对四种核苷酸中的一种严格成立；否则的话，F 称为不可延伸的。

定义 $\varepsilon(C, F, m)$ 为

$$\varepsilon(C,\ F,\ m)=\begin{cases}0, & \text{如果 } F \text{ 关于阵列 } C \text{ 在 } m \text{ 个核苷酸后是可延伸的}\\ 1, & \text{否则}\end{cases}$$

分枝概率 $p(C, n, m)$ 是一个在第 m 个核苷酸后不可延伸的随机 n 序列利用阵列 C 重构的概率，即

$$p(C,n,m) = \frac{1}{4^n}\sum_{F}\varepsilon(C,F,m)$$

其中求和是取遍所有长度为 n 的 4^n 条序列 F。

令 m 固定，且记 $p(C, n)=p(C, n, m)$。显然，$p(C, n)$ 是关于 n 的渐增函数。对一个给定的概率 p，满足条件 $p(C, n)\leqslant p$ 的最大 n 是允许用低于 p 的分枝概率进行明确的延伸的最大序列长度 $n_{\max}(C, p)$。对均匀阵列证明了 $n_{\max}(C,p)\approx\frac{1}{3}|C|p$，而二元阵则为 $n_{\max}(C,p)\approx\frac{1}{\sqrt{12}}|C|\sqrt{p}$。因此新的阵列与均匀阵列比较在最大序列长度上提高了$\sqrt{\frac{3}{4p}}$。相对于均匀阵列的 $n_{\max}\approx 210$，对 $p=0.01$ 的二元 64kb 阵列则有 $n_{\max}\approx 1\,800$。

5.11 多探针阵列与均匀阵列的比较

考虑序列 $F=X_1\cdots X_{m-1}X_mX_{m+1}\cdots X_n$ 且假设它的前缀 $F_m=X_1\cdots X_m$ 已经被重构。记 F_m 中的最后 $(l-1)$ 元组为 $V=X_{m-l+2}\cdots X_m$。为了简单起见，假设 $l\leqslant m$ 及 $l\ll n\ll 4^l=|C(l)|$。

利用 $C(l)$，序列 F 在 m 位之后是不可延伸的，如果谱 $F_{C(l)}$ 包含一个 VY_i 的 l 元组（这里 Y_i 是与 X_{m+1} 不同的任意核苷酸）。因此，$p(C(l),n)=1-P\{VY_1, VY_2, VY_3\notin F_{C(l)}\}$。假定在序列 F 给定的位置上找到 4^l 个 l 元组中的每一个的概率为$\frac{1}{4^l}$。F 的谱不包含 VY_1 的概率可粗略地估计为$\left(1-\frac{1}{4^l}\right)^{n-l+1}$（忽略潜在的自身重叠[269]及边缘影响）。$F$ 的谱既不包含 VY_1，也不包含 VY_2 和 VY_3 的概率可以估计为$\left[\left(1-\frac{1}{4^l}\right)^{n-l+1}\right]^3$。因此

$$\begin{aligned}P[C(l),n]&=1-P\{VY_1,VY_2,VY_3\notin F_{C(l)}\}\\&\approx 1-\left[\left(1-\frac{1}{4^l}\right)^{n-l+1}\right]^3\approx\frac{3n}{4^l}=\frac{3n}{|C(l)|}\end{aligned}\tag{5.2}$$

因此

$$n_{\max}(C(l),p)\approx\frac{1}{3}|C(l)|p$$

现在，估计 $p(C_{\text{bin}}(l),n)$，当 $n\ll\|C_{\text{bin}}(l)\|$。将前缀 F_m 中的最后的 l 元组记为 $V=X_{m-l+1}$，…，X_m，并且令 V_{WS} 和 V_{RY} 为 V，分别记入 $\{W, S\}$ 和 $\{R, Y\}$。在这种情况下，重构中的模糊性增加，谱 $F_{C_{\text{bin}}(l)}$ 既包含 $V_{WS}Y_i$ 多探针，也包含 $V_{RY}Y_i$ 多探针，当 $Y_i\neq X_{m+1}$。假设在 F 的一个给定位置从 $C_{\text{bin}}(l)$ 中找出多

探针的概率为$\frac{1}{4\times 2^l}$，且忽略自身重叠（self-overlaps）。则 F 的谱不包含 $V_{WS}Y_1$ 的概率可粗略地估计为$\left(1-\frac{1}{4\cdot 2^l}\right)^{n-l}$。因此 F 谱既包含 $V_{WS}Y_1$ 又包含 $V_{RY}Y_1$ 的概率为

$$\left[1-\left(1-\frac{1}{4\cdot 2^l}\right)^{n-l}\right]\left[1-\left(1-\frac{1}{4\cdot 2^l}\right)^{n-l}\right]\approx\frac{n^2}{4\cdot 2^l\cdot 4\cdot 2^l}$$

类似于式（5.2），可以推得如下式子

$$\begin{aligned}p[C_{\text{bin}}(l),n]&=P\{V_{WS}Y_i\in F_{C_{\text{bin}}(l)},V_{RY}Y_i\in F_{C_{\text{bin}}(l)}\}\\&\approx 1-\left(1-\frac{n^2}{4\cdot 2^l\cdot 4\cdot 2^l}\right)^3\approx 3\cdot\frac{n}{4\cdot 2^l}\cdot\frac{n}{4\cdot 2^l}=\frac{12n^2}{|C_{\text{bin}}(l)|^2}\end{aligned}$$

因此，对 $C_{\text{bin}}(l)$

$$n_{\max}[C_{\text{bin}}(l),p]\approx\frac{1}{\sqrt{12}}|C_{\text{bin}}(l)|\sqrt{p}$$

下面来估计缺口阵列 $C_{\text{gap}}(l)$ 的分枝概率。设 $m\geqslant 2l-1$，$n\ll\|C_{\text{gap}}(l)\|$。记 $U=X_{m-2l+4}\cdots X_{m-l+2}$。在这种情况下，当谱 $F_{C_{\text{gap}}(l)}$ 既包含 VY_i 的 l 多探针又包含 $U\underbrace{XX\cdots X}_{l-1}Y_i$ 的（$2l-1$）多探针（这里，$Y_i\neq X_{m+1}$）时模糊性增加。假定在 F 的一个给定位置从 $C_{\text{gap}}(l)$ 中找出每个多探针的概率为$\frac{1}{4^l}$，且忽略自身重叠。则 F 的谱不包含 VY_1 的概率可粗略地估计为$\left(1-\frac{1}{4^l}\right)^{n-l}$。因此，$F$ 的谱不包含 $U\underbrace{XX\cdots X}_{l-1}Y_1$ 概率可粗略地估计为$\left(1-\frac{1}{4^l}\right)^{n-(2l-1)+1}$。于是，$F$ 的谱既包含 VY_1 又包含 $U\underbrace{XX\cdots X}_{l-1}Y_1$ 的概率是

$$\left[1-\left(1-\frac{1}{4^l}\right)^{n-l}\right]\left[1-\left(1-\frac{1}{4^l}\right)^{n-2l+2}\right]\approx\frac{n^2}{4^l\cdot 4^l}$$

类似于式（5.2），可以推得如下式子

$$\begin{aligned}p[C_{\text{gap}}(l),n]&=P\{VY_i\in F_{C_{\text{gap}}(l)},UY_i\in S[C_{\text{gap}}(l),F]\}\\&\approx 1-\left(1-\frac{n^2}{4^l\cdot 4^l}\right)^3\approx 3\,\frac{n}{4^l}\cdot\frac{n}{4^l}=\frac{12n^2}{|C_{\text{gap}}(l)|^2}\end{aligned}$$

相似的论证可得

$$n_{\max}[C_{\text{alt}}(l),p]\approx\frac{1}{\sqrt{12}}|C_{\text{alt}}(l)|\sqrt{p}$$

5.12 DNA 阵列的制造

DNA 阵列可以使用大尺度的固定化聚合物合成技术 VLSIPS（very large

scale immobilized polymer synthesis）来制造[109,110]。在 VLSIPS 中，经过包含一系列化学步骤的照相平版印刷过程，探针每次增长一个核苷酸。每个核苷酸携带一个防止探针进一步增长的对光不稳的保护基团。这个基团通过用光照射探针可予删除。在每个化学步骤中，阵列中预先确定的区域被照射，于是从这个区域删除了对光不稳的保护基团，且为进一步的核苷酸增长“激活”了探针。从而整个阵列就易于接受一种特殊的核苷酸（它可以忍受自己的对光不稳的保护基团），但是反应只发生在激活的区域。每次这个步骤被重复，一个新的区域被激活，且单个核苷酸被添加到这个区域中的每个探针上。通过添加核苷酸到适当的序列中的合适区域，就可能在经过 $4l$ 步后增长出一个长度为 l 的完整的探针集合。光导合成可以任意达到阵列的所有位置，从而可被用来制作在任何位置具有任何探针的阵列。

合适的区域通过对阵列的一系列掩模照射来激活，如图 5.12 所示。一个掩模的黑色区域相应于被照射的阵列区域，而白色区域则相应于盲区。不幸的是，由于衍射、内部反射和散射，接近照射区域和掩模区域边界的点经常会受无意识的照射。在这些区域，很难确认核苷酸是否被添加。这个不确定性会使探针具有未知序列和长度，由于探针会和靶 DNA 链杂交，因此使实验数据的解释更复杂。人们已经在探索边界长度最小化的方法，以使不确定性的程度减小。下面介绍的二维 Gray 编码是最优化的 VLSIPS 掩模，它使所有掩模的总边界长度最小化。

如图 5.12 所示具有不同的 3 元组排列及为合成第一个核苷酸 A 的掩模的两个 $C(3)$ 阵列（只有第一个核苷酸为 A 的探针才显示）。图 5.12 下部的掩模的边界长度比图 5.12 上部的掩模的边界长度显著地小。试着在阵列 $C(l)$ 上安排探针，使得所有 $4l$ 掩模的总边界长度为最小。设 x 和 y 是两个 l 元组，记 $\delta(x, y)$ 为 x 和 y 不相同的位置数目。显然，所有掩模的总边界长度为 $2\sum\delta(x, y)$，和号取遍阵列上所有相邻探针对。这个观察建立起了边界长度的最小化与 Gray 编码之间的联系。

一个 l 位的 Gray 编码被定义为 0 和 2^l-1 之间的二进制数的一个置换，它使得相邻数仅差 1 位，像开始数和结束数。例如，4 位二进制反射 Gray 编码如下所示。

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0

Gray 编码可以由递归产生，可以由如下的 1 位 Gray 编码开始

$$G_1=\{0, 1\}$$

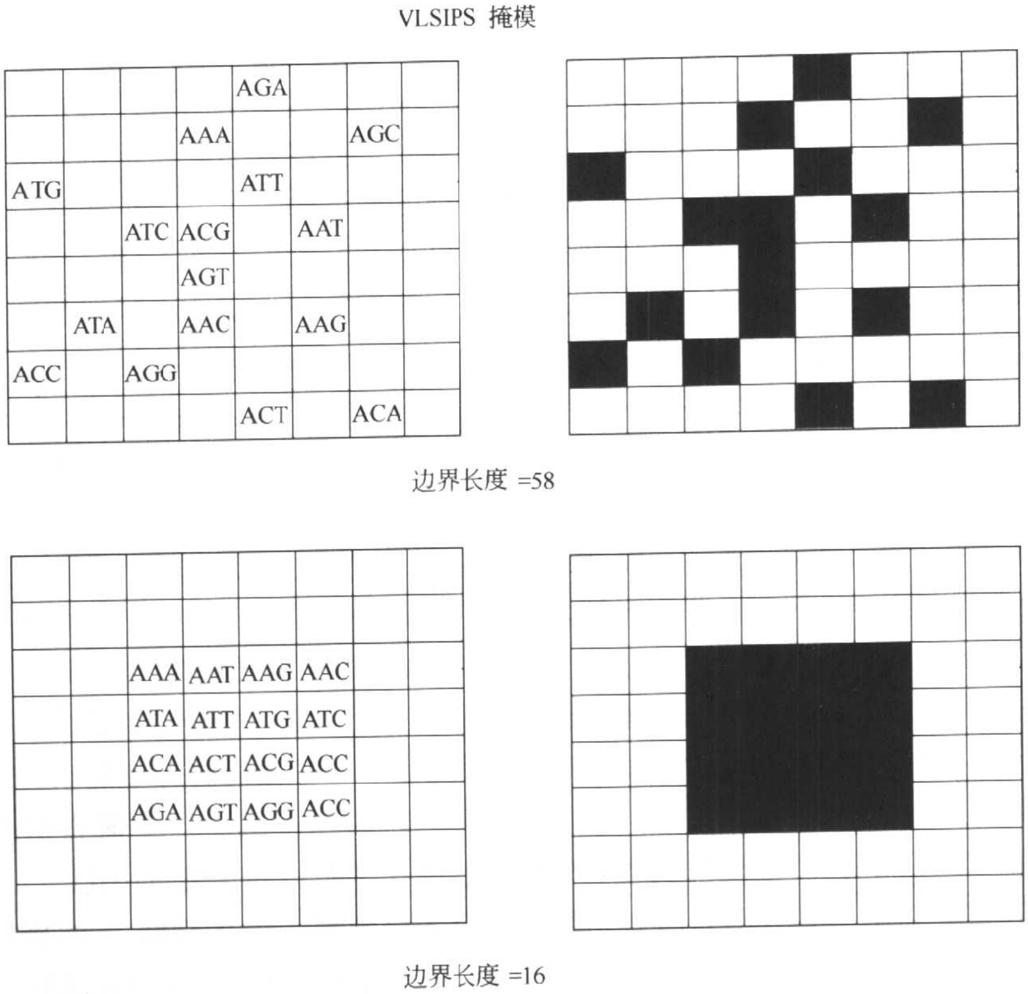


图 5.12 两个不同边界长度的掩模

对一个 l 位的 Gray 编码

$$G_l=\{g_1,g_2,\cdots,g_{2^l-1},g_{2^l}\}$$

定义一个（$l+1$）位 Gray 编码如下

$$G_{l+1}=\{0g_1,0g_2,\cdots,0g_{2^l-1},0g_{2^l},1g_{2^l},1g_{2^l-1},\cdots,1g_2,1g_1\}$$

将 G_l 中的元素简单复制并在元素前都添加上 0，将 G_l 中的元素简单复制并颠倒排序，再在元素前都添加上 1。显然，G_{l+1} 中的所有元素是不同的，G_{l+1} 中的相邻元素间仅差一位。

对覆盖 4 个字母的长度为 l 的字符串组成的二维 Gray 编码感兴趣。换句话说，希望产生一个 2^l 乘 2^l 阶矩阵，在此矩阵中，4^l 个 l 元组的每一个被安排于一个位置（i，j），并且每对邻近的 l 元组（水平或垂直）只相差一个位置。这样的 Gray 编码可由如下的一位数二维 Gray 编码产生

$$G_1=\begin{vmatrix}A & T\\ G & C\end{vmatrix}$$

如下，由 l 位数二维 Gray 编码

$$G_l = \begin{vmatrix} g_{1,1} & \cdots & g_{1,2^l} \\ \cdots & \cdots & \cdots \\ g_{2^l,1} & \cdots & g_{2^l,2^l} \end{vmatrix}$$

定义（$l+1$）位数二维 Gray 编码为

$$G_{l+1} = \begin{vmatrix} Ag_{1,1} & \cdots & Ag_{1,2^l} & Tg_{1,2^l} & \cdots & Tg_{1,1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ Ag_{2^l,1} & \cdots & Ag_{2^l,2^l} & Tg_{2^l,2^l} & \cdots & Tg_{2^l,1} \\ Gg_{2^l,1} & \cdots & Gg_{2^l,2^l} & Cg_{2^l,2^l} & \cdots & Cg_{2^l,1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ Gg_{1,1} & \cdots & Gg_{1,2^l} & Cg_{1,2^l} & \cdots & Cg_{1,1} \end{vmatrix}$$

特别

$$G_2 = \begin{vmatrix} AA & AT & TT & TA \\ AG & AC & TC & TG \\ GG & GC & CC & CG \\ GA & GT & CT & CA \end{vmatrix}$$

G_l 中的元素复制到 G_{l+1} 的左上象限，然后垂直和水平反射到邻接的三个象限。A、T、C 和 G 分别添加到上左，上右，下左，下右象限中的元素的前面。

上面的构造是许多可能的 Gray 编码中的一种。二维 Gray 编码可以通过二维乘积由任何一对一维 Gray 编码 G_1 和 G_2 产生 $G(i,j)=G_1(i)*G_2(j)$，其中 $*$ 是一个滑移算子（shuffle operation）（一个任意确定的滑移好像卡片的两面）。最简单的滑移是 $G_1(i)$ 和 $G_2(j)$ 的串联。对均匀阵列，Gray 编码的掩模在所有掩模中有最小的总边界长度，Gray 编码掩模的边界长度和标准的掩模的边界长度之比接近 $\frac{1}{2}$[99]。

5.13 若干其他问题和方法

5.13.1 利用通用碱基的 SBH

Preparata 等[280]进一步推进了多探针阵列的设想，介绍了这种阵列，对于一条长度为 n 的任意字符串的明确的重构所需要的探针数目，它们已达到了信息论的较低的界限。这些阵列利用了通用碱基，例如次黄嘌呤核苷（inosine），如果没有结合，则可正确地堆积，且在探针中扮演一个“不在乎”（don't care）符号的角色。这些阵列的设计与缺口阵列的设计相似，而这种缺口阵列具有更精心制

作的缺口模式。另一种 SBH 组合方法是由 Hubbell[171]提出的。

5.13.2 自适应 SBH

自适应 SBH 的概念可以由以下的例子来解释。想像一个超级程序员，他执行非常复杂的软件来计算巨大的数。在超级 PC 机上第一次运行后，他发现在他的程序中的一个循环里有个瓶颈，要算出这个巨大的数将需要一百年。在软件业中克服这个问题的通常方法是分析这个耗费时间的循环，定位这个瓶颈，然后编码出新的更快速的软件（或者是修改旧的软件）。然而，如果超级程序员薪酬很贵（现在是他的天下!），这可能不是最好的方法。一个不同的方法是分析这个耗时的循环，并且制作新的硬件使其能够更快地执行带瓶颈的循环，从而可以利用已有的软件计算巨大的数。当然，只有当制造新的超级 PC 的代价比雇佣超级程序员的花费便宜，这才是有意义的。

对自适应 SBH 而言，DNA 阵列是计算机，而 DNA 序列则是程序。不可随意修改程序（DNA 片段），但当知道了序列重构中的瓶颈后，就可以制造新的阵列。Skiena 和 Sundaram[316] 及 Margaritis 和 Skiena[232] 研究了无误差自适应 SBH，并且找到了序列重构所需测序次数的精致的理论界限。这个思想被 Kruglyak[210] 进一步发展了。

5.13.3 SBH-类型的鸟枪测序法

Idury 和 Waterman[175] 建议在传统的 DNA 测序中对序列组装使用欧拉路 SBH 方法。这个主意既简单又优秀：对足够大的 l（即 $l=30$），将每个序列长度 n 的检读处理为 $n-l+1$ 个 l 元组。因为在人类基因组中大多数的 30 元组是惟一的，这种方法在无误差数据情况中导出一个非常有效的测序算法。Idury 和 Waterman[175] 也试图将此算法改编成适应有测序误差的情况。

5.13.4 DNA 阵列的保真探针

在 DNA 阵列制造中，目前的质量控制方法是合成小型测试探针集，用它探测制作过程中的变化。这些保真探针是由探阵的相同的拷贝组成，但是，它们是在阵列制作过程中的不同步骤故意地制作的。一个已知的靶和这些探针杂交，杂交的结果反映了制作过程中的质量。它并不只探测变化，而且分析出现的变化，指出制作过程中的哪一步出了错。Hubbell 和 Pevzner[172] 介绍一个组合的方法，构建一个小型的保真探针集，它不仅探测变化，而且指出错误的制作步骤。

6 序列比较

6.1 引言

DNA 中的突变是一个自然的进化过程：DNA 复制错误引起核苷酸的替换、插入和删除，导致 DNA 文本的“编辑”。DNA 序列之间的相似性可以作为有共同进化起源的线索（例如人和黑猩猩的球蛋白基因之间的相似性），又或者是具有共同功能的线索（例如 *v-sys* 致癌基因和生长激素之间的相似性）。

建立致癌基因和涉及正常生长发育的基因之间的联系[89,353]是序列比较中首次成功的事例。致癌基因是病毒中一种能使受感染细胞产生癌变的基因。在猴肉瘤（simian sarcoma）病毒中的致癌基因 *v-sys* 能引起细胞的失控增长从而使猴子患上癌症。而表面上与它毫无相关的生长因子 PDGF（血小板生长因子）则是促进细胞生长的一种蛋白质。当把这两种基因进行对比时，却发现了显著的相似性。这个发现证实了一个猜想，那就是癌症可以由一种正常的生长基因在不适当的时间发生转变而引起。

Levenshtein[219]在 1966 年引入了“编辑距离”的概念，作为把一条字符串转变成另一条字符串所需要的最小编辑操作数，这里的编辑操作是指插入一个符号，删除一个符号或把一个符号替换成另外一个符号。大多数 DNA 序列比较算法仍使用这种或与这稍有差别的操作集。Levenshtein 虽然引入了编辑距离的定义但却从没有描述一个找出两条序列之间的编辑距离的算法。从语音处理[347]到分子生物学[251]的不同应用领域，这个算法被发现和再发现了许多次。尽管在各个不同应用上算法的细节稍有不同，但它们都是动态规划算法的变形。

找出序列之间的差异（编辑距离）常常等价于找出序列之间的相似之处。例如，如果编辑操作仅限于插入和删除（没有替换），则编辑距离问题等价于最长共同子序列（longest common subsequence，LCS）问题。数学家早在用于序列比较的动态规划算法发现之前就已经对 LCS 问题感兴趣了。对对称群的研究揭示了表示论与在两置换间寻找 LCS 的问题之间令人惊奇的联系。对于这个问题的第一个算法是由 Robinson[287]于 1938 年提出的。Robinson 的工作却被遗忘了，直到 20 世纪 60 年代以后，Schensted[306]和 Knuth[201]又重新发现了 LCS 和杨氏表之间的这个关系。

尽管序列比较算法的多数方面都被 LCS 问题抓住了，生物学家还是愿意使用联配（alignment）这个词来指称 DNA 和蛋白质序列比较。字符串 V 和 W 之

间的联配是一个两行矩阵，使得第一行（第二行）依序容纳着 $V(W)$ 中的字符，其间散布着一些空格。联配的得分定义为它的各列得分之和。各列得分通常对于一致的字符取正对于差异较大的字符取负。

在早期有关序列联配的论文里，科学家们试图找到整个字符串 V 和 W 之间的相似性，即全局联配。这对于同一蛋白质家族成员之间的比较是有意义的，比如球蛋白，在从果蝇到人的各种生物体中都非常保守并具有几乎相同的长度。在许多生物学应用中，V 和 W 的子序列之间的联配得分可能都大于整个序列之间的联配得分。这就是有名的局部联配问题。举个例子，homeobox（同源异型域）基因是调节胚胎发育的基因，在很大范围的物种中都存在。尽管 homeobox 基因在不同的物种中有很大不同，但它们中有一个区域叫 homeodomain（同源异型域）的却是非常保守的。这里就有了个问题，那就是如何找到这个保守区域和忽略相似程度很低的区域。Smith 和 Waterman[320] 于 1981 年提出了动态规划算法的一个聪明的修正方法从而解决了这个局部联配问题。

回到 1983 年，那时去找致癌基因和涉及正常生长发育的基因之间的相似是令人惊讶的。而在今天，不把一条新测序的基因放到巨大的 GenBank 数据库中去寻找它的相似序列那倒更令人奇怪了。然而，现在在 GenBank 数据库中搜寻已不能像 20 年前那般容易了。当试图在一个 10^9 大小的数据库中寻找一条 10^3 长的基因的最相近匹配时，即使二次动态规划算法也许都太慢了。一个方法是使用快速并行硬件来执行联配算法；另一个方法就是使用快速的启发式算法，它实行起来很好但不能保证找到最佳匹配。

分子生物学中许多用于快速数据搜索的启发式算法都采用了相同的过滤思想（filtering idea）。过滤是以观察为基础的，一个好的联配通常包括相同的或非常相似的片段。这样人们就可以先搜索这些短的子序列并利用它们作为进一步分析的种子。这种快速序列比较的过滤思想要追溯到 20 世纪 70 年代早期，比 FASTA 和 BLAST 算法的发明还要早。Knuth[202] 于 1973 年提出了一个基于观察的只有一次错配的模式匹配算法，即字符串只有单个错配，而在前半段或后半段必须是严格匹配的。例如，一个 9 元组❶只有一次错配的近似模式匹配可以简化为 4 元组的精确模式匹配，对其再加以扩展就又变成 9 元组的近似匹配了。这就为过滤掉不具有这共同的 4 元组的字符串的位置提供了机会，从而就有一大部分被滤掉了。计算分子生物学中的这种过滤思想最初是由 Dumas 和 Ninio[92] 于 1982 年提出的，更为重要的是 Wilbur 和 Lipman[368]、Lipman 和 Pearson[225] 在他们的 FASTA 算法中采用了它。后来又在 BLAST 程序中得到了进一步发展，而 BLAST 程序正是当今分子生物学中的主流数据库搜索工具[5]。

❶ 9 元组，是指由 9 个字符组成的短字符串——译者注

6.2 最长共同子序列问题

定义字符串 $V=v_1\cdots v_n$ 和 $W=w_1\cdots w_m$ 的共同子序列为指标序列

$$1\leqslant i_1<\cdots<i_k\leqslant n$$

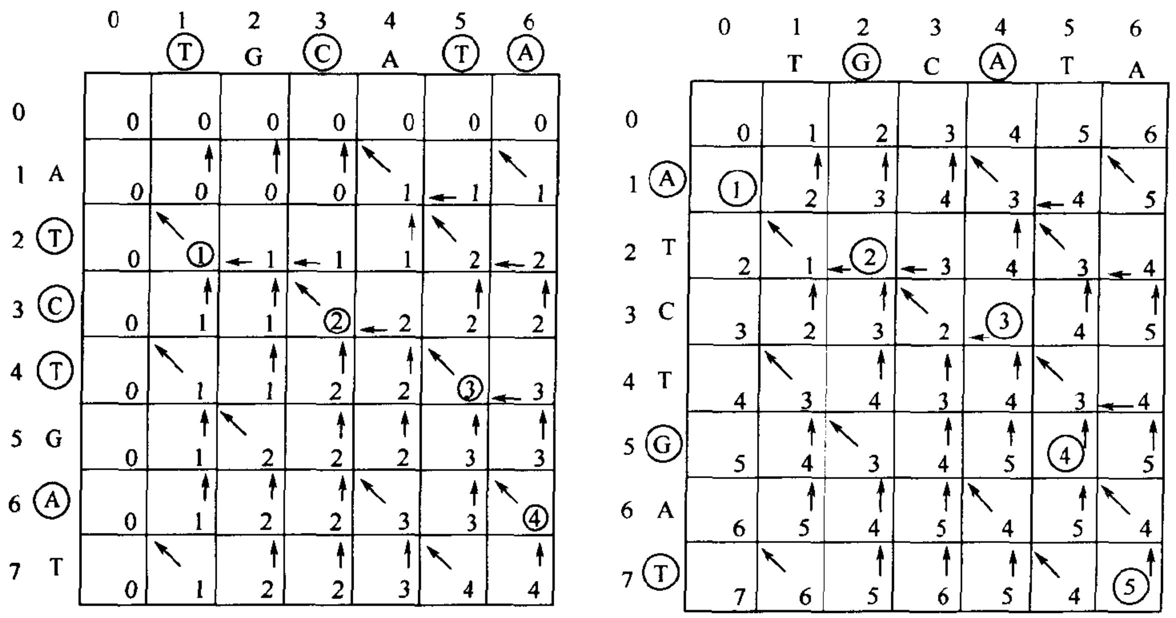
和指标序列

$$1\leqslant j_1<\cdots<j_k\leqslant m$$

使得

$$v_{i_t}=w_{j_t} \text{ 对 } 1\leqslant t\leqslant k$$

令 $s(V, W)$ 为 V 和 W 的一个最长共同子序列（LCS）的长度（k）。很明显，$d(V,W)=n+m-2s(V,W)$ 就是 V 转变成 W 的最小插入和删除数。图 6.1 显示了字符串 V=ATCTGAT 和 W=TGCATA 的一段长为 4 的 LCS 以及把 V 转变成 W 所需的 2 次插入和 3 次删除后的一段最短序列。



(a) 计算相似度 $s(V, W)=4$
V 和 W 有共同子序列 TCTA

(b) 计算距离 $d(V, W)=5$
在 V 中删除 A、G、T 并插入 C 可以转变为 W

```
A T - C - T G A T
- T G C A T - A -
```

(c) 联配

图 6.1 计算最长共同子序列的动态规划算法

计算 $s(V, W)$ 的简单动态规划算法曾被许多作者独立地发现过。令 $s_{i,j}$ 为 V 的前 i 个字符 $V_i=v_1\cdots v_i$ 组成的子字符串和 W 的前 j 个字符 $W_j=w_1\cdots w_j$ 组成的子字符串之间的 LCS 的长度。令 $s_{i,0}=s_{0,j}=0$ 对所有 $1\leqslant i\leqslant n$ 和 $1\leqslant j\leqslant m$。则 $s_{i,j}$ 可如下递归计算

$$s_{i,j}=\max\begin{cases}s_{i-1,j}\\ s_{i,j-1}\\ s_{i-1,j-1}+1 & \text{if } v_i=w_j\end{cases}$$

第一项（第二项）对应 $v_i(w_j)$ 不在 V_i 和 W_j 的 LCS 中的情况，第三项对应于 v_i 和 w_j 都在 V_i 和 W_j 的 LCS 中的情形（v_i 匹配 w_j）。图 6.1（a）中的动态规划表显示了 V 和 W 之间的相似得分 $s(V, W)$ 的计算过程，而图 6.1（b）中的表则显示了 V 和 W 之间的编辑距离的计算过程。编辑距离 $d(V, W)$ 是依照初始条件 $d_{i,0}=i$，$d_{0,j}=j$，对所有的 $1\leqslant i\leqslant n$ 和 $1\leqslant j\leqslant m$ 和如下的递归公式进行计算

$$d_{i,j}=\min\begin{cases}d_{i-1,j}+1\\ d_{i,j-1}+1\\ d_{i-1,j-1} & \text{if } v_i=w_j\end{cases}$$

V 与 W 间的 LCS 长度可以从动态规划表中的元素(n,m)读到。为了构建一个 LCS，必须记录下 $s_{i,j}$ 在上述递归计算中的最大值，即 3 个量($s_{i-1,j}$，$s_{i,j-1}$，或 $s_{i-1,j-1}+1$)其中之一，然后再通过动态规划表的回溯就行了。下面的算法实现了这个目标，它引入了 3 个箭头←、↑和↖来分别表示上面的 3 种情况。

LCS(*V*,*W*)
 for $i\leftarrow 1$ **to** n
  $s_{i,0}\leftarrow 0$
 for $i\leftarrow 1$ **to** m
  $s_{0,i}\leftarrow 0$
 for $i\leftarrow 1$ **to** n
  for $j\leftarrow 1$ **to** m
   if $v_i=w_j$
    $s_{i,j}\leftarrow s_{i-1,j-1}+1$
    $b_{i,j}\leftarrow$ ↖
  else if $s_{i-1,j}\geqslant s_{i,j-1}$
    $s_{i,j}\leftarrow s_{i-1,j}$
    $b_{i,j}\leftarrow$ ↑
  else
    $s_{i,j}\leftarrow s_{i,j-1}$
    $b_{i,j}\leftarrow$ ←
 return s and b

下面的递归程序依照箭头←、↑和↖指向的从(n,m)到$(0,0)$的路径回溯可印制出最大共同子序列。初始调用是 **PRINT-LCS**(b,V,n,m)。

```
PRINT-LCS(b,V,i,j)
if i=0 or j=0
    return
if b_{i,j}=↖
    PRINT-LCS(b,V,i-1,j-1)
    print v_i
else if b_{i,j}=↑
    PRINT-LCS(b,V,i-1,j)
else
    PRINT-LCS(b,V,i,j-1)
```

6.3 序列联配

令 $\mathcal{A}$ 为一有 k 种字母的字符集，而 V 和 W 则为由 $\mathcal{A}$ 中元素组成的两条序列。又令 $\mathcal{A}'=\mathcal{A}\cup\{-\}$ 为一扩展字符集，这里“—”代表空格。字符串 $V=v_1,\cdots,v_n$ 和 $W=w_1,\cdots,w_m$ 的一个联配就是一个 $2\times l$ 的矩阵 $A(l\geqslant n,m)$，使得 A 的第一行（第二行）依序包含了 $V(W)$ 的字符，其间并散布着 $l-n$ $(l-m)$ 个空格[见图 6.1(c)]。假定联配矩阵中没有那一列包含两个空格。联配中包含一个空格的列被称为一个插缺（indel），空格在第一行（第二行）的叫做插入（缺失）。两行中字母都相同的列叫做匹配，而字母不相同的列叫做错配。包含扩展字符 x 和 y 的列的得分由一个 $(k+1)\times(k+1)$ 阶的矩阵规定，它定义了 $\mathcal{A}'$ 中每两个字符 x 和 y 的相似性得分 $\delta(x,y)$。全体联配的得分就被定义为它的各列得分之和。最简单的矩阵 δ 就是规定匹配时得分为 $\delta(x,x)=1$，每个错配的罚分为 $\delta(x,y)=-\mu$，每个插入或缺失为 $\delta(x,-)=\delta(-,x)=-\sigma$。这时联配的得分定义为 # 匹配个数 $-\mu$ # 错配个数 $-\sigma$ # 插缺个数。最长共同子序列问题是参数为 $\mu=\infty$，$\sigma=0$ 的联配问题。用于蛋白质序列比较的通用矩阵，PAM（point accepted mutations），可接受点突

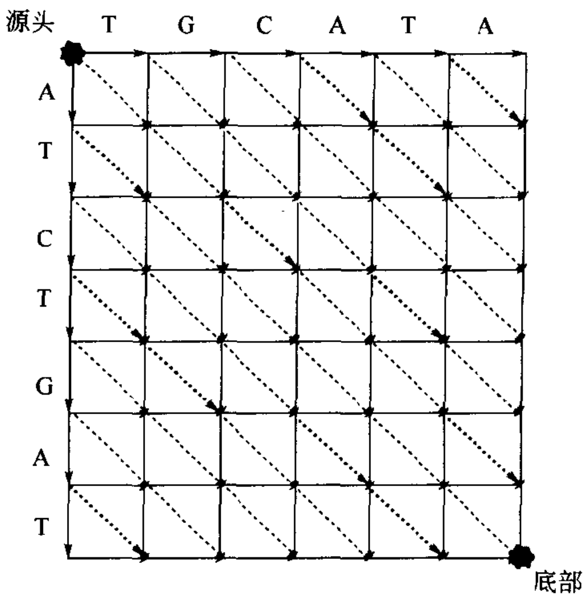


图 6.2 编辑图。插入和缺失边的权重是 $-\sigma$，错配边的权重是 $-\mu$，匹配边的权重是 1

↓ 缺失； ⟶ 插入；
↘ 错配； ⋱ 匹配

变矩阵和 BLOSUM，则反映了相关序列在进化中氨基酸 x 替换氨基酸 y 的频率[6,82,158]。

（全局）序列联配问题是要找到序列 V 和 W 的具有最大得分的联配。相应的求 V_i 和 W_j 的最优联配得分 $s_{i,j}$ 的递归公式为

$$s_{i,j}=\max\begin{cases}s_{i-1,j}+\delta(v_i,-)\\ s_{i,j-1}+\delta(-,w_j)\\ s_{i-1,j-1}+\delta(v_i,w_j)\end{cases}$$

V 和 W 的每一联配都对应于 V 和 W 序列的编辑图(edit graph)中的一条路径（见图 6.2）。因此，序列联配问题对应于有向无圈图中从源点到汇点的最长路径问题。

6.4 局部序列联配

生物学上有意义的相似经常只是出现在 DNA 片段的某些部分，而在其他地方则不出现。在这种情况下生物学家试图求得 $s(v_i\cdots v_{i'},w_j\cdots w_{j'})$ 的最大值，这个最大值是从 V 和 W 的所有子字符串 $v_i\cdots v_{i'}$ 和 $w_j\cdots w_{j'}$ 中求得的。

全局联配问题相应于在编辑图中寻找顶点$(0,0)$和(n,m)之间最长路径，而局部联配问题则相应于在编辑图中任意顶点(i,j)和(i',j')之间的所有路径中寻找一条最长的路径。对于这个问题的一个简单直接但低效的方法就是找出每一对顶点(i,j)和 (i',j') 之间的最长路径。替代寻找从任一顶点(i,j)出发的最长路径，局部联配问题可简化为在从某一特定源顶点出发的路径中寻找最长路径，从源顶点到其他每一顶点加一权值为 0 的边。这些边为从源顶点到任何其他顶点(i,j)提供了一个“自由的”跨越。下列递归公式中的一点小差异反映了编辑图中的这种改进[320]

$$s_{i,j}=\max\begin{cases}0\\ s_{i-1,j}+\delta(v_i,-)\\ s_{i,j-1}+\delta(-,w_j)\\ s_{i-1,j-1}+\delta(v_i,w_j)\end{cases}$$

$s_{i,j}$的最大值就代表了 V 和 W 之间局部联配的得分（而不是全局联配的得分 $s_{n,m}$）。

最优局部联配仅报告了编辑图中的最长路径。同时，好几个局部联配可能都有生物学意义，上述方法就是在寻找 k 个最好的非重叠局部联配[168,359]。这些方法对于具有相似区域的多功能域蛋白质之间的比较尤其重要，这种相似区域出现在相互对比的蛋白质中的位置被打乱了。在这种情况下，单单一个局部联配是不可能揭示所有有意义的相似性。

6.5 有缺口罚分的联配

突变通常都是 DNA 复制过程中出错的表现。自然界经常删除或插入作为一个单元的整个子序列，而删除或插入单个核苷酸则与此相反。联配中的缺口就是各单行序列中的一段连续的空格。很自然的，可认为一段有 x 个空格的缺口的得分并不仅仅是 x 次插缺的得分之和，而是另有相当的功用。由于设了仿射缺口罚分（affine gap penalties），一段长为 x 的缺口的得分是 $-(\rho+\sigma x)$，这里 $\rho>0$是引入缺口的罚分，$\sigma>0$ 则是缺口中每个符号的罚分。仿射缺口罚分可同编辑图中长的垂直边或水平边相对应［例如，一条从(i,j)到$(i+x,j)$的边长$-(\rho+\sigma x)$］，通过进一步计算图中的最长路径也可得到说明。由于编辑图中仿射缺口罚分的边数增加，乍一看联配算法的运行时间似乎要从 $O(n^2)$增长到 $O(n^3)$，然而，下列 3 个递归公式却能使运行时间降下来

$$\overset{\downarrow}{s}_{i,j}=\max\begin{cases}\overset{\downarrow}{s}_{i-1,j}-\sigma\\ s_{i-1,j}-(\rho+\sigma)\end{cases}$$

$$\vec{s}_{i,j}=\max\begin{cases}\vec{s}_{i,j-1}-\sigma\\ s_{i,j-1}-(\rho+\sigma)\end{cases}$$

$$s_{i,j}=\max\begin{cases}s_{i-1,j-1}+\delta(v_i,w_j)\\ \overset{\downarrow}{s}_{i,j}\\ \vec{s}_{i,j}\end{cases}$$

变量 $\overset{\downarrow}{s}_{i,j}$计算 V_i 与 W_i 之间的联配的得分，以一个缺失而结束（即 W 中有一个缺口），而变量 $\vec{s}_{i,j}$计算联配的得分则以一个插入而结束（即 V 中有一个空格）。$\overset{\downarrow}{s}_{i,j}$和 $\vec{s}_{i,j}$的递归公式中的第一项对应于延伸缺口，而第二项则对应于起始缺口。尽管仿射缺口罚分是当今最常用的模型，由一些研究表明非线性缺口罚分也许比仿射缺口罚分更具有优势[133,356]。以非线性缺口罚分用于联配的有效算法由 Miller 和 Myers[236]，还有 Galil 和 Giancarlo[116]提出过。

6.6 空间效率高的序列联配

在比较长的 DNA 序列片段时，序列联配中的有限资源是空间而不是时间。Hirschberg[163]提出了一个分而治之的方案，它可在线性空间中进行联配而只需花费两倍的计算时间。

对于序列联配的动态规划算法的时间复杂度大概是编辑图中的边数，即 $O(nm)$。空间复杂度大概就是编辑图中的顶点数，即 $O(nm)$。然而，如果只想计算出联配的得分（而不是联配本身），那么空间复杂度可减小为仅仅两倍于编辑图的一个单列中的顶点数，即 $O(n)$。这个减少来自观测到计算联配得分 $s_{*,j}$

(j 列) 所惟一需要的值就是联配得分 $s_{*,j-1}$($j-1$ 列)。因此，在计算 $j,j+1,\cdots$ 列联配得分时 $j-1$ 列前的联配得分就可以丢掉了。恰恰相反，计算联配（即找到编辑图中的最长路径）需要回溯整个矩阵($s_{i,j}$)，因此，整个回溯矩阵都需要被存储下来，这就导致了 $O(nm)$的空间需求。

编辑图中的最长路径连接起始点(0,0)和下降点(n,m)并通过一个（未知的）中间点$\left(i,\frac{m}{2}\right)$（简单计，假设 m 是偶数)。试着找到它的中间点而不是整条最长路径。在线性空间中，这可以通过计算得分 $s_{*,\frac{m}{2}}$ [从(0,0)到$\left(i,\frac{m}{2}\right)$的最长路径的长度，其中 $0\leqslant i\leqslant n$] 和从$\left(i,\frac{m}{2}\right)$到$(n,m)$的路径的得分而做到。后者得分可以计算倒转的编辑图（即所有边的方向都颠倒的图）中从(n,m)到$\left(i,\frac{m}{2}\right)$的路径的得分 $s^{\text{reverse}}_{*,\frac{m}{2}}$ 而得到。$s_{i,\frac{m}{2}}+s^{\text{reverse}}_{i,\frac{m}{2}}$就是从(0,0)到$(n,m)$并通过顶点$\left(i,\frac{m}{2}\right)$的最长路径的长度。因此，$\max_i\left(s_{i,\frac{m}{2}}+s^{\text{reverse}}_{i,\frac{m}{2}}\right)$就计算出了最长路径的长度并确定了一个中间顶点。

计算这些值需要的时间等于左方框区域（从第 1 列到第$\frac{m}{2}$列）加上右方框区域（从第$\frac{m}{2}+1$ 列到第 m 列)，空间复杂度为 $O(n)$（见图 6.3)。当中间顶点$\left(i,\frac{m}{2}\right)$找到之后，求从(0,0)到$(n,m)$的最长路径的问题就可以分成两个子问题：求从(0,0)到中间点$\left(i,\frac{m}{2}\right)$的最长路径和求从中间点$\left(i,\frac{m}{2}\right)$到$(n,m)$的最长路径。取代这些路径的寻找，试着去找各相应方框中的中间顶点（见图 6.3)。这可以在等于这些方框区域的时间内做到，它要比初始方框区域小两倍。如此计算下去，将会在 $area+\frac{area}{2}+\frac{area}{4}+\cdots\leqslant 2\times area$ 的时间内找出所有方框的中点[1]，从而可在时间复杂度为 $O(nm)$和空间复杂度为 $O(n)$内算出最长路径。

Path(*source*, *sink*)
if *source* and *sink* 是在相连的列
　　输出从 *source* 到 *sink* 的最长路径
else
　　middle←在 *source* 和 *sink* 之间的中间顶点

[1] 此处 *area* 指方框区域面积——译者注

Path(*source*, *middle*)

Path(*middle*, *sink*)

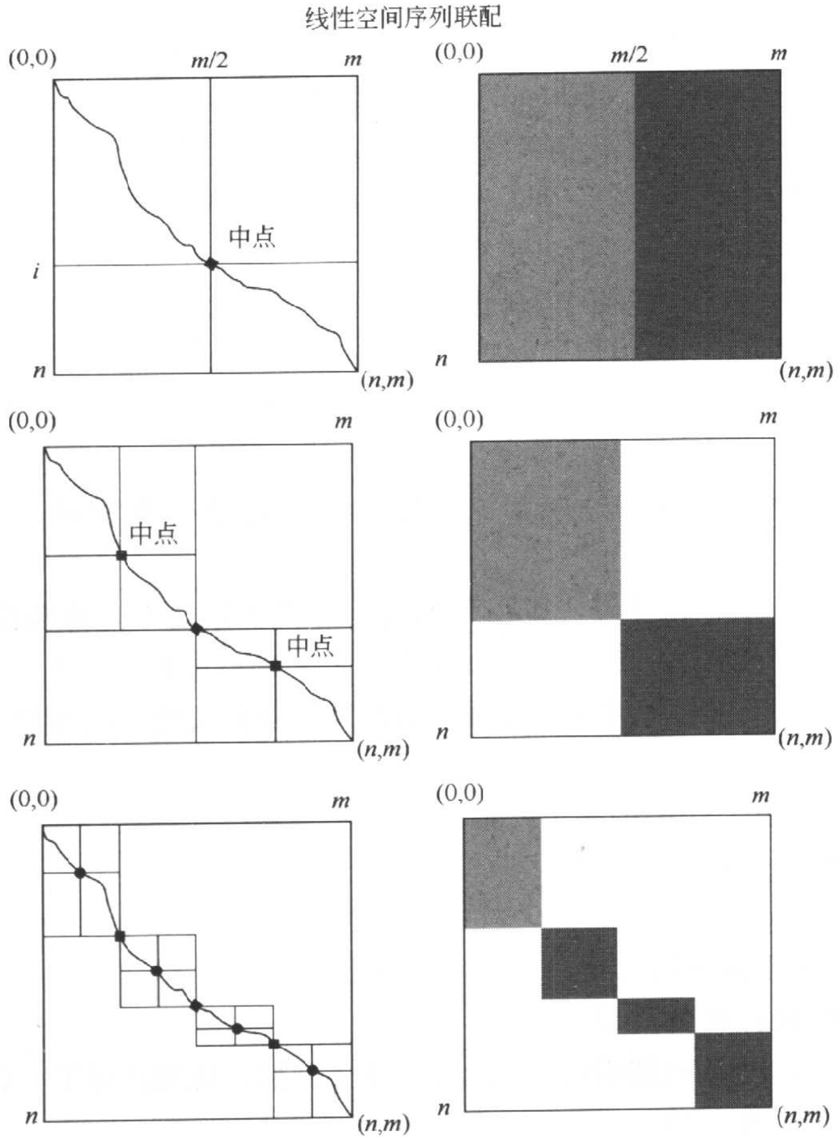


图 6.3 空间效率高的序列联配

6.7 杨氏表

排列 $\pi = x_1 x_2 \cdots x_n$ 的一个递增子序列是一条使 $x_{i_1} < x_{i_2} \cdots < x_{i_k}$ 的下标序列 $1 \leqslant i_1 < \cdots i_k \leqslant n$。递减序列也类似定义。找最长递增子序列（longest increasing subsequence，LIS）等价于在 π 和一致排列 $12\cdots n$ 之间找 LCS。大家都知道，n 个元素的每一个排列都有一条至少长为$\sqrt{n}$的或递增或递减的子序列。这个结论与下面将要介绍的有关 LCS 的非动态规划方法紧密相关。

整数 n 的一个划分是这样一个由正整数 $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_l$ 并使得 $\sum_{i=1}^{i=l} \lambda_i = n$ 组

成的序列。如果 $\lambda=(\lambda_1,\lambda_2,\cdots,\lambda_l)$ 是 n 的一个划分，记作 $\lambda\vdash n$。设 $\lambda=(\lambda_1,\lambda_2,\cdots,\lambda_l)\vdash n$，那么杨氏图表或形状（diagram 或 shape）λ，就是一个有 n 个格子的阵列，它有 l 个左移位的行使得第 i 行就包含 λ_i 个格子，其中 $1\leqslant i\leqslant l$。图 6.4（a）表示了 $(4,2,1,1)\vdash 8$ 的杨氏图表，而图 6.4（d）表示了 $(4,2,2,1)\vdash 9$ 的杨氏图表。一个杨氏表（tableau）（或者简单地说是形状 λ 的表）是通过用数字 $1,2,\cdots,n$ 一一对应地取代杨氏图表 λ 而得到的一个阵列 [见图 6.4（d）]。如果它的行或列是递增序列，一个 λ 表是标准的。Sagan[292] 极好地介绍了关于杨氏表的组合数学，以下参照这本书进行叙述。

一个双表是一对具有相同的杨氏图表 λ 的标准杨氏表。Robinson-Schensted-Knuth（RSK）算法明确描述了 n 个格子的双表和对称群 S_n（n 个元素的所有排列）之间的一个双射关系。它最初是由 Robinson[287] 提出的（一个相当模糊的术语）与表示论相联系。Schensted 在 1961 年[306] 又重新发现，并在纯组合数学基础上清楚地表述了这个算法。后来，Knuth[201] 把这个算法推广用于 LCS 的情况。

令 P 表示一个不完整表，即行和列增加的格子数是不同的杨氏图表。对一个阵列 R 和一个元素 x，定义 R_{x+} 为 R 中大于 x 的最小元素，R_{x-} 为 R 中小于 x 的最大元素。对于不在 P 中 x，依据下面的算法来定义把 x 插入 P 的行插入。

$R\leftarrow P$ 的第一行
While x 小于行 R 的某个元素
    在 R 中用 x 替换 R_{x+}
    $x\leftarrow R_{x+}$
    $R\leftarrow$ 下面一行
把 x 放在行 R 的最后

将行插入 x 进 P 的结果记作 $r_x(P)$。注意到插入规则确保了 $r_x(P)$ 仍然有增长的行和列（见图 6.4）。

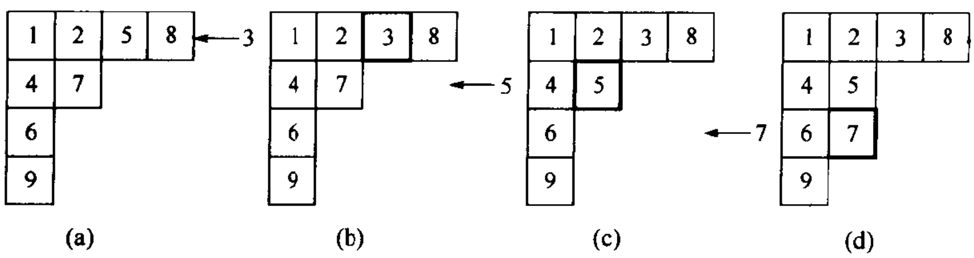


图 6.4 行插入 3 到杨氏表

排列和双表之间的双射记作 $\pi\xrightarrow{\text{RSK}}(P,Q)$，这里 $\pi\in S_n$，P，Q 是标准 λ 表，$\lambda\vdash n$。对于一个排列 $\pi=x_1\cdots x_n$，构造表的序列

$$(P_0,Q_0)=(\varnothing,\varnothing),(P_1,Q_1),\cdots,(P_n,Q_n)=(P,Q)$$

这里 x_1，…，x_n 是插入到各 P 中的，而 1，…，n 则放在各 Q 中，使得对所有的 i，P_i 的形状与 Q_i 的形状相一致。

放置一个元素进表中甚至比插入还容易。设 Q 是一个不完整表，(i,j)是 Q 的一个外拐角单元。如果 k 大于 Q 中的每一个元素，那么要把 k 放置在 Q 的(i,j)格子中，只需设 $Q_{i,j}=k$(见图 6.5)。

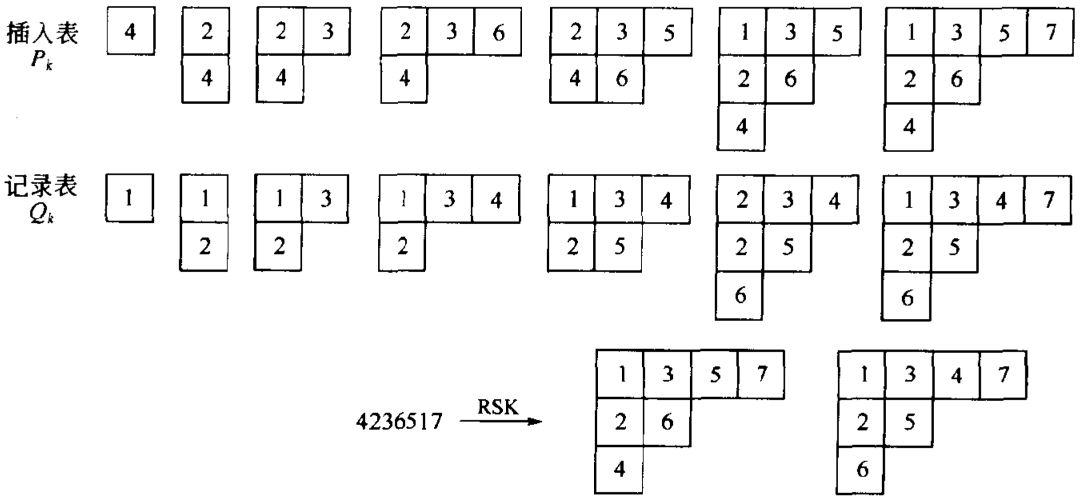


图 6.5 排列 $\pi=4236517$ 的 RSK 算法

最后，叙述怎样从 $\pi=x_1\cdots x_n$ 建立双表（P_i，Q_i）的序列。假设(P_{k-1},Q_{k-1})已经被构建，定义 $P_k=r_{x_k}(P_{k-1})$，而 Q_k 为把 k 放入 Q_{k-1}的格子（i，j）后的结果，该格子为插入终止处（见图 6.5）。把 P 叫做插入表，把 Q 叫做记录表。

如果给定图 6.4 (a) 中的图表 $R_x(P)$ 和最后要加入的元素 (7) 的位置，能重新构造图 6.4 (d) 中的图表吗？因为元素 7 被 $R\,7--=5$ 从前一行中撞了出来（见 RSK 算法），能重构图 6.4 (b) 中的图表。因为元素 5 被元素 $R\,5--=3$ 从第一行撞了出来，能重构初始表 P。这个直观过程隐含了下面的 RSK 定理。

定理 6.1 映射 $\pi\xrightarrow{\text{RSK}}(P,Q)$是一个在 S_n 的元素和具有同样形状 $\lambda\vdash n$ 的标准表对偶元之间的一个双射。

证明 通过一步步倒转 RSK 算法构造一个逆双射$(P,Q)\xrightarrow{\text{RSK}}\pi$。从定义$(P_n,Q_n)=(P,Q)$开始。假设$(P_k,Q_k)$已经构建，将找到 x_k（π 的第 k 个元素）和(P_{k-1},Q_{k-1})，找到 Q_k 中放着的 k 格子(i,j)。由于这是 Q_k 中的最大的元素，$P_{i,j}$必须是在构造 P_k 过程中被转移的最后面的一个元素。可以利用下列程序来从 P 中删除 $P_{i,j}$。为了方便起见，假定在 P_k 的第一行上面有一个空的第 0 行存在。

令 $x\leftarrow P_{i,j}$并清除 $P_{i,j}$

$R \leftarrow P_k$ 的第$(i-1)$行

While R 不是 P_k 的第 0 行

  用 R 中的 x 替换 R_{x-}

  $x \leftarrow R_{x-}$

  $R \leftarrow$接下来的一行

$x_k \leftarrow x$

很容易看到，P_k 经过刚才叙述的删除处理之后就是 P_{k-1}，而且 Q_k 在 k 去掉之后就是 Q_{k-1}。继续这个过程，最终将以倒序覆盖 π 的所有元素。■

引理 6.1 如果 $\pi = x_1 \cdots x_n$，且 x_k 进入 P_{k-1} 的第 j 列，那么以 x_k 结尾的 π 的最长递增子序列的长度为 j。

证明 对 k 使用归纳法。当 $k=1$ 时结果是显然的，因此假设它对于直到 $k-1$ 的所有值都成立。

首先，我们需要证明有以 x_k 结尾的长为 j 的增长子序列的存在。令 y 为 P_{k-1} 的 $(1, j-1)$ 格子中的元素。则有 $y < x_k$，因为 x_k 是从第 j 列进入的。同样，利用归纳法，可知存在一个以 y 结尾、长为 $j-1$ 的增长子序列。把这个子序列同 x_k 合并就得到了想要的长为 j 的子序列。

现在再证明不存在一个以 x_k 结尾的更长的增长子序列。如果存在这样一条子序列，令 x_i 为 x_k 前面的一个元素。由归纳法，x_i 插入第 j 列右边的某列。则有 P_i 的格子$(1, j)$中的元素 y 满足 $y \leqslant x_i < x_k$。然而，在表的给定位置中的项目是不会随着子序列的插入而增长的（见 RSK 算法）。这样 P_{k-1} 的格子 $(1, j)$ 中的元素就比 x_k 小，这同 x_k 替代它的事实相矛盾。■

这个引理蕴含着下列内容。

定理 6.2 排列 π 的最大递增子序列的长度就是 $P(\pi)$第一行的长度。

6.8 最长共同子序列的平均长度

令 $\mathcal{V}$ 和 $\mathcal{W}$ 为两个由定义于同一字母表上的 n 个字母构成的字符串的集合。给定 $\mathcal{V}$，$\mathcal{W}$ 和一个 $\mathcal{V} \times \mathcal{W}$ 上的概率测度 p，考虑 LCS 的平均长度，它等于

$$s(n) = \sum_{V \in \mathcal{V}, W \in \mathcal{W}} s(V, W) p(V, W)$$

这里 $s(V, W)$为 V 和 W 之间的 LCS 的长度。LCS 平均长度问题中有两个例子是尤其让人感兴趣的。

随机排列中的最长递增子序列 $s_{\text{per}}(n)$。$\mathcal{V}$ 仅包含字符串$(1, \cdots, n)$，$\mathcal{W}$ 包含长度为 n 的所有排列，且 $p(V, W) = \frac{1}{n!}$。寻找 $s_{\text{per}}(n)$的问题是由 Ulam[341] 在 1961 年提出的。Hammersley[150] 在 1972 年证明了

$$\lim_{n\to\infty}\frac{s_{per}(n)}{\sqrt{n}}=s_{per}$$

其中 s_{per} 是一个常数。甚至在收敛性被证明之前，Baer 和 Brock[15] 在 1968 年在广泛计算的基础上就已猜想到 $s_{per}=2$。Hammersley[150] 证明了 $\frac{\pi}{2}\leqslant s_{per}\leqslant e$。后来，Kingman[198] 把界限改进为 $1.59<c<2.49$。Logan 和 Shepp[227] 以及 Vershik 和 Kerov[342] 通过对随机杨氏表的非常具有技巧挑战性的渐进分析以及利用定理 6.1 和定理 6.2 证明了 $s_{per}=2$。

k 元字母表上的最长共同子序列 $s_k(n)$。在 k 元字母表上，$\mathcal{V}$ 和 $\mathcal{W}$ 都包含所有的 k^n 个 n 元字母组合，并有 $p(V,W)=\frac{1}{k^n k^n}$。Chvátal 和 Sankoff[70] 注意到

$$\lim_{n\to\infty}\frac{s_k(n)}{n}=s_k$$

这里 s_k 是一个常数。他们给出了 s_k 的上下界，后又由 Deken[83,84] 和 Chvátal 及 Sankoff[71] 进行了改进。

在 20 世纪 80 年代，两个关于 s_k 的猜想被陈述如下。

Sankoff-Mainville 猜想[305] $\lim\limits_{k\to\infty}(s_k\sqrt{k})=2$

Arratia-Steele 猜想[328] $s_k=\frac{2}{1+\sqrt{k}}$

这两个猜想作为杨氏表第一行的长度的公式化阐述。取代求出第一行的长度，可试着求出能同时产生杨氏表，尤其是第一行的长度的所有特征的杨氏表的有限形状。乍一看，这个问题似乎比直接求第一行的长度更一般，更困难。然而，对 $s_{per}=2$ 的证明却揭示了一个有点荒谬的情形：那就是求整个杨氏表的有限形状比使用特定的组合/概率参数去寻找某一特定行的期望长度要容易。特别是，Ulam 问题的特定的组合解仍是未知的。

最近，在最长递增子序列的研究上有了积极的进展。Aldous 和 Diaconis[2] 利用交互质点表示法（通过在一条由均匀分布在一个区间中的 n 个独立随机实数构成的序列上求 LIS 来建立模型）给出了 $s_{per}=2$ 的又一个不同证明。Baik 等[21] 证明了 $s_{per}(n)=2\sqrt{n}-\mu n^{1/6}+o(n^{1/6})$，这里 $\mu=1.711\cdots$。

令 Λ_n 为有 n 个格子的所有形状的集合。记一个对应着排列 π 的插入表 P 为 $P(\pi)$，考虑排列集合 $\Gamma=\{\pi:P(\pi)$第一行中包含 $n\}$。对于一个给定的有 n 个格子的形状 λ，令 $\Gamma_\lambda=\{\pi:P(\pi)$第一行包含 n，并且 $P(\pi)$有形状 $\lambda\}$。对于一个随机排列 $\pi\in S_n$，令 p_n 为 $P(\pi)$第一行中包含 n 的概率。

引理 6.2 $p_n\leqslant\frac{1}{\sqrt{n}}$

证明 如果λ是一个形状，以λ_+代表在λ的第一行末尾增加一个新的格子而派生出来的形状。观察到n在第一行的形状$\mu \in \Lambda_n$的标准表的数目等于形状$\lambda \in \Lambda_{n-1}$的标准表的数目，其中$\lambda_+ = \mu$。令$f_\lambda$是形状为$\lambda$的杨氏表数目。根据RSK定理

$$|\Gamma| = \sum_{\mu \in \Lambda_n} |\Gamma_\mu| = \sum_{\lambda \in \Lambda_{n-1}} f_\lambda \cdot f_{\lambda_+}$$

这里f_λ是形状为λ的杨氏表块数。它暗含着以下内容

$$p_n = \frac{|\Gamma|}{n!} = \sum_{\lambda \in \Lambda_{n-1}} \frac{f_\lambda \cdot f_\lambda}{n!} = \sum_{\lambda \in \Lambda_{n-1}} \frac{(n-1)!}{n!} \frac{f_\lambda \cdot f_{\lambda_-}}{f_\lambda \cdot f_\lambda} \frac{f_\lambda \cdot f_\lambda}{(n-1)!}$$

$$= \sum_{\lambda \in \Lambda_{n-1}} \frac{1}{n} \frac{f_{\lambda_+}}{f_\lambda} p(\lambda)$$

按照RSK定理，$p(\lambda) = \frac{f_\lambda \cdot f_\lambda}{(n-1)!}$是对应于形状$\lambda$的一个随机排列$\pi \in S_{n-1}$的概率。记$E(X) = p_n$，为随机变量$X = \frac{1}{n} \frac{f_{\lambda_+}}{f_\lambda}$的平均值。应用不等式$E(X) \leqslant \sqrt{E(X^2)}$，可得

$$p_n^2 \leqslant \sum_{\lambda \in \Lambda_{n-1}} \frac{1}{n \cdot n} \frac{f_{\lambda_+} \cdot f_{\lambda_+}}{f_\lambda \cdot f_\lambda} \frac{f_\lambda \cdot f_\lambda}{(n-1)!} = \sum_{\lambda \in \Lambda_{n-1}} \frac{1}{n \cdot n!} f_{\lambda_-} \cdot f_{\lambda_+}$$

$$= \frac{1}{n} \sum_{\lambda \in \Lambda_{n-1}} \frac{f_{\lambda_+} \cdot f_{\lambda_+}}{n!} = \frac{1}{n} \sum_{\lambda \in \Lambda_{n-1}} p(\lambda_+) \leqslant \frac{1}{n} \sum_{\mu \in \Lambda_n} p(\mu) \leqslant \frac{1}{n}$$

因为λ_+的变动范围覆盖了所有第一行长度大于第二行长度的$\mu \in \Lambda_n$。 ■

下面的定理是由Vershik和Kerov在20世纪70年代后期证明的（显然是从未公开发表过）。第一个公开发表的证明在后来才出现[276]。

定理 6.3 $s_{\text{per}} \leqslant 2$

证明 给定$\pi \in S_n$，令$p_k(n)$为元素k出现在$P(\pi)$的第一行的概率。注意到$p_k(n) = p_k$（S_n中一个随机排列的元素$1, \cdots, k$是均匀分布在所有可能的相关次序上的）。按照引理6.2，$P(\pi)$第一行的期望长度是

$$r_1 = \sum_{k=1}^{n} p_k(n) = \sum_{k=1}^{n} p_k \leqslant \sum_{k=1}^{n} \frac{1}{\sqrt{k}}$$

由于$\frac{1}{\sqrt{k}} \leqslant 2\left(\sqrt{k} - \sqrt{k-1}\right)$，得到$r_1 \leqslant 2\sqrt{n}$。因为$P(\pi)$的第一行的长度等于$\pi$的最长递增子序列的长度，故$s_{\text{per}} \leqslant 2$。 ■

6.9 广义序列联配和对偶性

一个半序集是一个偶对$(P, \prec)$，其中P是一个集合，$\prec$是P上的一个传递

的、非自反的二元关系，即，若 $p \prec q, q \prec r$ 则有 $p \prec r$。一条链 $p_1 \prec p_2 \cdots \prec p_t$ 是 P 的一个子集，其中任何两个元素都是可比较的，而一条反链是其中任何两个元素都不可比较的一个子集。半序 $\prec$ 和 $\prec^*$ 被称作共轭的，如果对任何两个不同的 $p_1, p_2 \in P$ 成立下面的条件

$$p_1 \text{ 和 } p_2 \text{ 是} \prec \text{可比的} \Leftrightarrow p_1 \text{ 和 } p_2 \text{ 是} \prec^* \text{不可比的}$$

所关注的是最长 $\prec$ 序列。即以 $\prec$ 关系存在的最长的链（广义序列联配）。令 $I=\{1,2,\cdots,n\}$，$J=\{1,2,\cdots,m\}$，$P \subset I \times J$。我们的兴趣在于两条序列 $V=v_1 v_2 \cdots v_n$ 和 $W=w_1 w_2 \cdots w_m$ 之间的对比，并有 $P=\{(i,j): v_i = w_j\}$。令 $p_1=(i_1, j_1)$，$p_2=(i_2, j_2)$ 为 $I \times J$ 中的任意两个元素。记

$$\Delta(p_1, p_2) = (\Delta i, \Delta j) = (i_2 - i_1, j_2 - j_1)$$

考虑 $I \times J$ 上半序的一些例子（如图 6.6 所示相应的链）。半序 $\prec_1$ 和 $\prec_4$ 以及半序 $\prec_2$ 和 $\prec_3$ 是共轭的。

- 公共子序列（CS）

  $p_1 \prec_1 p_2 \Leftrightarrow \Delta i > 0, \Delta j > 0$
- 公共森林（CF）

  $p_1 \prec_2 p_2 \Leftrightarrow \Delta i \geqslant 0, \Delta j \geqslant 0$
- 公共反向子序列（CIS）

  $p_1 \prec_3 p_2 \Leftrightarrow \Delta i > 0, \Delta j < 0$
- 公共反向森林（CIF）

  $p_1 \prec_4 p_2 \Leftrightarrow \Delta i \geqslant 0, \Delta j \leqslant 0$

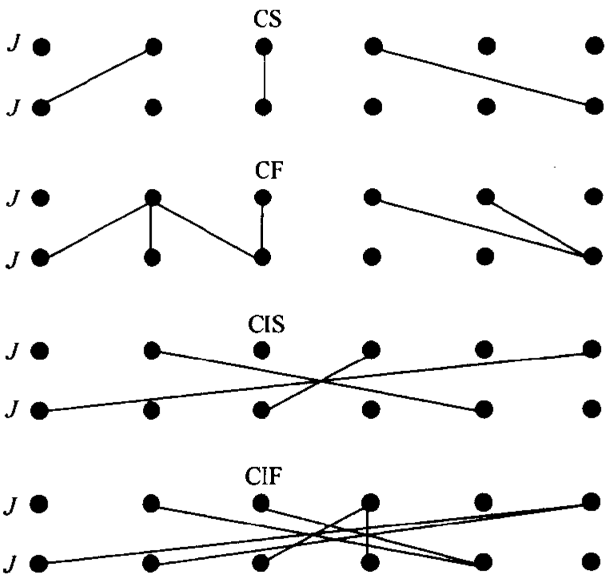


图 6.6 半序 CS，CF，CIS 和 CIF 中的关系链

令 $\mathcal{C}$ 为集合 P 的一簇子集。$\mathcal{C}' \subseteq \mathcal{C}$ 叫作 P 的一个覆盖，如果每一 $p \in P$ 被包含在至少一个子集 $C \in \mathcal{C}'$ 中。$\mathcal{C}'$ 中的元素的数目称为覆盖 $\mathcal{C}'$ 的规模，一个最小规模的覆盖称作 P 被 $\mathcal{C}$ 的最小覆盖。下面的定理是由 Dilworth[86] 证明的。

定理 6.4 设 P 是一个半序集。P 的被链最小覆盖的规模等于 P 中的一条最大反链的规模。

引理 6.3 设 $\prec$ 和 $\prec^*$ 为 P 上的共轭半序。那么 P 中一条最长 $\prec$ 序列的长度等于 P 被 $\prec^*$ 序列最小覆盖的规模。

证明 根据 Dilworth 定理，$\prec^*$ 的最长反链的长度等于 P 被 $\prec^*$ 链最小覆盖的规模。由于 $\prec$ 和 $\prec^*$ 是共轭的，$\prec^*$ 的每一反链就是 $\prec$ 的链，$\prec$ 的每一条链就是 $\prec^*$ 的一条反链。因此，P 中一条最长 $\prec$ 序列的长度等于 P 被 $\prec^*$ 序列最小覆盖的规模。 ■

因为 CS 和 CIF 表现了共轭半序关系，引理 6.3 还蕴涵了下面的定理[273]。

定理 6.5 最长 CS 的长度等于由 CIF 最小覆盖的规模。

考虑 P 上的一个二元关系，其定义为

$$p_1 \sqsubset p_2 \Leftrightarrow p_1 \prec p_2 \text{ 或 } p_1 \prec^* p_2$$

引理 6.4 $\sqsubset$ 是 P 上的一个线性序。

证明 首先证明 $p_1 \sqsubset p_2$，$p_2 \sqsubset p_3$ 意味着 $p_1 \sqsubset p_3$。如果 $p_1 \sqsubset p_2$ 且 $p_2 \sqsubset p_3$，那么下列条件之一将成立。

(1) $p_1 \prec p_2$ 且 $p_2 \prec p_3$。

(2) $p_1 \prec p_2$ 且 $p_2 \prec^* p_3$。

(3) $p_1 \prec^* p_2$ 且 $p_2 \prec p_3$。

(4) $p_1 \prec^* p_2$ 且 $p_2 \prec^* p_3$。

在情形 (1)，$p_1 \prec p_2$ 且 $p_2 \prec p_3$ 蕴涵着 $p_1 \prec p_3$，因此有 $p_1 \sqsubset p_3$。在情形 (2)，$p_1 \prec p_2$ 且 $p_2 \prec^* p_3$ 暗含着既没有 $p_3 \prec p_1$ 也没有 $p_3 \prec^* p_1$。(在第一种情形下，$p_3 \prec p_1$ 且 $p_1 \prec p_2$ 暗含着 $p_3 \prec p_2$，与 $p_2 \prec^* p_3$ 相矛盾。在第二种情形下，$p_2 \prec^* p_3$ 且 $p_3 \prec^* p_1$ 暗含着 $p_2 \prec^* p_1$，与 $p_1 \prec p_2$ 相矛盾)。因此 $p_1 \prec p_3$ 或 $p_1 \prec^* p_3$，就暗含着 $p_1 \sqsubset p_3$。注意到情形 (3) 和情形 (4) 分别同情形 (2) 和情形 (1) 是对称的，因此已经证明了关系 $\sqsubset$ 是传递性的。这个引理是从对每一对 p_1，p_2 的观察中得来的，即要么 $p_1 \sqsubset p_2$，要么 $p_2 \sqsubset p_1$。 ■

6.10 序列比较的原始对偶方法

定理 6.5 揭示了 LCS 和最小覆盖问题之间的一个关系，并导致了利用最小覆盖寻找 LCS 的想法的产生。下面介绍能同时解决广义联配和最小覆盖问题的一个非动态规划算法[273]。

令 $\mathcal{P}=p_1p_2\cdots p_l$ 为 P 中元素的一个任意排序，并且 $\mathcal{P}_i=p_1p_2\cdots p_i$。令 $\mathcal{C}_i=\{C_1,C_2,\cdots,C_j\}$是 $\mathcal{P}_i$ 被$\prec^*$ 序列的一个覆盖，在令 $p_1^{\max}$，$p_2^{\max}$，…，$p_j^{\max}$ 分别对应为 C_1，C_2，…，C_j 中的 $\prec^*$ 最大元。令 k 为达到下面条件的最小下标$(1\leqslant k\leqslant j)$

$$p_k^{\max} \prec^* p_{i+1} \tag{6.1}$$

如果条件式 (6.1) 对所有的 k 都不满足，则置 $k=j+1$。为方便起见，定义 C_{j+1}为一个空集。COVER 算法通过增加 p_{i+1} 到 C_k 从 $\mathcal{C}_i$ 构造了一个覆盖 $\mathcal{C}_{i+1}$。如果 $k<j+1$，COVER 算法就使 C_k 扩大

$$\mathcal{C}_{i+1}=\{C_1,C_2,\cdots,C_{k-1},C_k\cup\{p_{i+1}\},C_{k+1},\cdots,C_j\}$$

如果 $k=j+1$，COVER 算法就把$\{p_{i+1}\}$作为一个新的$\prec^*$序列增加到覆盖 $\mathcal{C}_{i+1}$ 中

$$\mathcal{C}_{i+1}=\{C_1,C_2,\cdots,C_j,C_{j+1}=\{p_{i+1}\}\}$$

这个算法同样也保留了一个回溯

$$b(p_{i+1})=\begin{cases} p_{k-1}^{\max} & \text{如果 } k>1 \\ \varnothing & \text{否则} \end{cases}$$

从一个空覆盖 $\mathcal{C}_0$ 开始，COVER 算法在迭代 l 次后建立了 P 的一个覆盖 $\mathcal{C}_l$，且覆盖的规模由 P 的排序决定。覆盖 $\mathcal{C}_l$ 的规模就是最长≺序列的长度的上界。下面的定理说明了如果 $\mathcal{P}$ 是 P 在 $\sqsubset$ 中的排序，那么 COVER 算法就是能同时解决最长≺序列问题和由≺* 序列最小覆盖的问题的原始对偶算法。

定理 6.6 如果 $\mathcal{P}=p_1p_2\cdots p_l$ 是 P 依序 $\sqsubset$ 的一个排序，那么 COVER 算法将由≺* 序列建立 P 的一个最小覆盖 $\mathcal{C}_l=\{C_1, C_2, \cdots, C_t\}$。而回溯 $b(p)$就为每一个 $p\in C_t$ 定义了一个长为 t 的最长≺序列。

证明 对每一个 $i(1\leqslant i\leqslant l)$，覆盖 $\mathcal{C}_i=\{C_1,C_2,\cdots,C_j\}$满足下面的条件

$$\forall k>1, \forall p\in C_k : b(p)\prec p \tag{6.2}$$

很明显，这个条件对 $\mathcal{C}_1$ 是成立的。假设它对 $\mathcal{C}_i$ 成立，然后证明对 $\mathcal{C}_{i+1}$也成立。分两种情况考虑。

• 情况 1 $k<j+1$[条件式(6.1)]，在这种情况下 $b(p_{i+1})=p_{k-1}^{\max}$。因为 $\mathcal{P}$ 是 $\sqsubset$-排序，$p_{k-1}^{\max}\sqsubset p_{i+1}$，因此要么有 $p_{k-1}^{\max}\prec p_{i+1}$或者有 $p_{k-1}^{\max}\prec^* p_{i+1}$。因为 k 是达到 $p_k^{\max}\prec^* p_{i+1}$，$p_{k-1}^{\max}\prec^* p_{i+1}$的最小下标，因此条件式（6.2）对 $\mathcal{C}_{i+1}$成立。

• 情况 2 $k=j+1$，在这种情况下 $b(p_{i+1})=p_j^{\max}$。因为 $\mathcal{P}$ 是 $\sqsubset$ 排序的，因此要么有 $p_j^{\max}\prec p_{i+1}$要么有 $p_j^{\max}\prec^* p_{i+1}$。因为 $k=j+1$，条件式（6.1）对每一个 $k\leqslant j$ 都不成立。因此，$p_j^{\max}\prec p_{i+1}$，且条件式（6.2）对 $\mathcal{C}_{i+1}$成立。

很显然，每一个满足条件式（6.2）的覆盖 $\mathcal{C}_l=\{C_1, C_2, \cdots C_t\}$ 都对每个 $p\in C_t$（经过回溯）决定了一个长为 t 的≺序列。根据引理 6.3，每一条这样的序列就是一条≺最长序列，而 $\mathcal{C}_l$ 就是≺* 序列对 P 的一个最小覆盖。 ■

对于 n 个字母的序列之间的 LCS，COVER 算法可以在 $O(nL)$时间内实现，这里 L 是最大共同子序列的长度，或者在 $O((l+n)\lg n)$时间内完成，其中 l 是两条序列之间的全部碱基配对数。对于寻找 LCS 的经典动态规划算法的改进后来又由 Hirschberg[164] 以及 Hunt 和 Szymanski[174] 提出来。实际上，Hirschberg 算法和 Hunt-Szymanski 算法都可以看成带多种数据结构的 COVER 算法的实现。COVER 算法中的≺* 链对应于 Hirschberg 算法中的 k 候选项。COVER 算法中≺* 链的最大元素对应于 Apostolico[9] 算法中的优势匹配，它是 Hunt-Szymanski 算法的改进。

6.11 序列联配和整数规划

LCS 的对偶与 Reinert 等[283] 提出的用于序列比较的一个新的多面体方法是紧密相关的。他们把联配问题表述为一个整数线性规划，并且据报道这个方法用

于多序列联配可解决的情况要超过现行的动态规划方法。下面来介绍 LCS 问题和整数规划之间的关系。

设 P 为一个有着半序$\prec$的集合，并令$\prec^*$为一个共轭半序。设 x_e 为与 $e\in P$ 相关连的权。在序列 $v_1\cdots v_n$ 和 $w_1\cdots w_m$ 的 LCS 问题中，P 为位置对 $e=(i,j)$ 的集合，使得 $v_i=w_j$，$x_e=1$，当且仅当位置 i 和 j 在 LCS 中是匹配的。于是，LCS 问题可以表述成下面的整数规划问题

$$\sum_{x_e\in\alpha} x_e\leqslant 1 \text{ 对 } P \text{ 中(最大的)反链 } \alpha$$

$$\sum_{e\in P} x_e\rightarrow \max$$

设 y_α 是一个与 P 中的最大反链 α 相关的变量。上述问题的对偶系统如下。

$$\sum_{e\in\alpha} y_\alpha\geqslant 1 \text{ 对每一个 } e\in P$$

$$\sum_{\alpha} y_\alpha\rightarrow \min$$

因为$\prec$的反链就是$\prec^*$的链，于是上述系统就变成表示共轭半序$\prec^*$的有向非循环图中的最小路径覆盖问题，而这个问题大家都知道存在一个整数解的[75]。

6.12 近似字符串匹配

有 k 个误配的近似字符串匹配涉及一条称为文本的字符串 $t_1\cdots t_n$，一条称为查询的短字符串 $q_1\cdots q_p$，和整数 k 及 m。查询匹配问题就是找出查询 $q_i\cdots q_{i+m-1}$ 所有长度为 m 的子字符串以及所有的最多有 k 个误配的文本 $t_1\cdots t_{j+m-1}$。在 $p=m$ 时，查询匹配问题就转化为有 k 个误配的近似字符串匹配问题。

有 k 个误配的近似字符串匹配问题在计算机科学中曾被专门研究过。当 $k=0$ 时，它就变为经典的序列匹配问题，其求解可在 $O(n)$ 时间内完成[45,203]。当 $k>0$时，对近似字符串匹配的简单强制算法所需的运行时间则为 $O(nm)$。近似字符串匹配的线性时间算法是由 Ivanov[177] 以及 Landau 和 Vishkin[213] 设计的。对一个固定大小的字母表，这些算法在最坏情况下的运行时间是 $O(kn)$。

尽管这些算法考虑了最好和最坏情形下的性能，但它们在实际操作中还远不是最好的方法[140]。因此，与最坏情形的运行时间相反，一些基于过滤的方法就更着重于预期运行时间[16,17,140,334,371]。

对近似字符串匹配使用过滤算法涉及两个步骤。第一步在文本中预选出一个与查询有潜在相似的位置的集合。第二步检验每一个潜在位置，排除多于 k 个误配的潜在匹配。把算法的第一步中找到的潜在匹配数记为 p。预选通常需在 $\alpha n+O(p)$时间内完成，这里 α 是一个小的常数。如果潜在匹配数很小，并且潜在匹配检验不太慢的话，这种方法可以使速度显著提高。

字符串匹配问题的过滤思想最初是由 Karp 和 Rabin[189] 对 $k=0$ 的情形提出

来的。对 $k>0$ 的情形，Owolabi 和 McGregor[258] 利用了 l 元组过滤的思想，它是基于这样一种简单观察，即如果一个查询近似匹配一个文本，那么对于足够大的 l，它们至少共有一个 l 元组。所有的查询和文本共有的 l 元组可以用散列法很容易地找到。如果共有的 l 元组数相对较小，它们可被检验，并且所有在 k 个误配以内的真实匹配就能很快地被定位。

l 元组过滤是基于下面简单的观察。

引理 6.5 如果字符串 $x_1 \cdots x_m$ 和字符串 $y_1 \cdots y_m$ 以至多 k 个误配而配对，那么对 $l=\left\lfloor \frac{m}{k+1} \right\rfloor$，它们就共有一个 l 元组，即有 $x_i \cdots x_{i+l-1}=y_j \cdots y_{j+l-1}$，对某些 $1 \leqslant i,\ j \leqslant m-l+1$。

这个引理促成了一种 l 元组过滤算法，它适用于带 k 个误配的查询匹配。

FILTRATION 算法 探测一个查询与一个文本之间的所有 m 匹配，直到有 k 个误配。

- **潜在匹配探测** 找到查询和文本中所有的 l 元组匹配，$l=\left\lfloor \frac{m}{k+1} \right\rfloor$。
- **潜在匹配检验** 通过向左向右延伸检验每一个潜在的匹配，直到或者第一次出现 $k+1$ 个误配，或者已到了查询或文本的两端。

引理 6.5 保证了 FILTRATION 过滤算法找到小于等于 k 个误配的长为 m 的所有匹配。过滤算法中的潜在匹配探测可以用散列法来实行。过滤算法的运行时间是 $\alpha n+O(pm)$，这里 p 是在算法第一步探测到的潜在匹配数，α 是一个小的常数。对于一个由 A 个等可能字符形成的 Bernoulli 文本来说，潜在匹配的期望数大致为 $E(p)=\frac{nq}{A^l}$，对大的 A 或 l 都可生成一个快速算法。

6.13 依靠数据库的序列比较

简单地说，序列 V 和 W 的点阵就是每一项为 0 或 1 的一个矩阵，其中在位置 (i,j) 为 1 就表示从 V 的第 i 个位置和 W 的第 j 个位置开始的 l 元组相重合。一个流行的蛋白质数据库搜索工具，FASTA[225]，就使用了 l 元组过滤，通常设置 $l=2$（依氨基酸字母表）。出现在两条字符串间的 l 元组位置形成了一个（隐含的）点阵，表示了这两条字符串之间的相似性。FASTA 进一步聚集该点阵中在同一对角线上的那些点，并试图把靠得近的对角线组在一起。

利用共有的 l 元组去发现相似性也有一些缺点。BLAST[5]，这个分子生物学中的主流数据库搜索工具使用了替换矩阵来改进（隐含的）点阵的结构，以加强对对角线的分析。实质上，它是想通过引入更加严格的规则来改进 FASTA 的过滤效率，以找到少一些但更好的潜在匹配。BLAST 的另一个特色就是利用了 Altschul-Dembo-Karlin 统计数据[85,186]来估计一些统计意义。对于任何两个 l 元

组 $x_1 \cdots x_l$ 和 $y_1 \cdots y_l$，BLAST 定义片段得分为 $\sum_{i=1}^{l} \delta(x_i, y_i)$，其中 $\delta(x,y)$ 为氨基酸 x 和氨基酸 y 之间的相似性得分。最大片段对（MSP）是在两条序列的所有片段对中有着最大得分的一对 l 元组。分子生物学家也许对全部保守序列都有兴趣，而不仅仅是那些有最高得分的序列。一个片段对是局部最大的，如果通过延长或缩短双方的片段它的得分并不增加的话。

BLAST 试图找出在查询序列和数据库中所有得分高出一个设定的阈值的局部最大片段对。阈值的选择是参考 Altschul-Dembo-Karlin 统计数据，它可使人确定出不太可能发生的片段对得分的最低值。BLAST 既给出片段得分高于阈值的序列，也给出一些片段得分没有过阈值但在组合上有统计意义的片段对。

BLAST 放弃了 l 元组过滤的思想而采用了一种与之不同的策略去寻找潜在匹配。它找出和查询中的 l 元组匹配得分超过阈值的所有 l 元组。这既可以以一种直接的方法去做——通过找出数据库中所有的近似于查询的子字符串的出现位置——也可以以一种间接的方法去做。例如，如果阈值足够高，那么这种字符串的集合就不会太大，则在数据库中就可以搜索出这些字符串的确切出现位置。这是一个研究得很透的组合模式匹配问题，快速 Aho 和 Corasick[1] 算法可用于确定数据库中所有这些字符串的出现位置。在潜在匹配被定位之后，BLAST 试图延伸它们看其最终得分是否高于阈值。Altschul 等[7] 通过允许插入和删除以及组合同一对角线或相近对角线上的匹配进一步改进了 BLAST。

6.14 多重过滤

对于有着 A 个等概率字符的 Bernoulli 文本，定义一个过滤算法的过滤效率为 $\frac{E(r)}{E(p)}$，即为带 k 个误配的期望匹配数 $E(r)$ 和潜在期望匹配数 $E(p)$ 之间的一个比值。例如，当 $k=1$ 时，l 元组过滤的效率 $\approx \frac{A-1}{A^{\lceil \frac{m}{2} \rceil}}$，且随着 m 和 A 增加而迅速减少。这就让人想到设计一种提高了过滤效率的过滤算法。效率比值越大，过滤算法中检验阶段的运行时间就越短。

从位置 4 开始的、缺口大小为 3、整体大小为 10 的带缺口 4 元组

Pevzner 和 Waterman[274] 介绍了一种算法，只要花费过滤时间线性增加的代价就能使潜在匹配数呈指数缩减。它以在探测阶段时间呈线性增长为代价，换得 FILTRATION 算法的检验阶段的时间显著减少，这是很有意义的。考虑到检验通常要比探测花费的时间长，因而这个技术为过滤参数的优化选择提供了一个折中方案。

位置 $i, i+t, i+2t, \cdots, i+jt, \cdots, i+(l-1)t$ 的集合叫做一个缺口大小为 t 以及整

体大小为 $1+t(l-1)$ 的带缺口 l 元组（见图 6.7）。连续 l 元组则可以被看成缺口大小为 1，整体大小为 l 的带缺口 l 元组。如果由一段模板序列和一段文本序列共有的 l 元组开始于模板序列的位置 i 和查询序列的位置 j，就把 (i,j) 叫做 l 元组的坐标。定义 l 元组坐标 (i_1,j_1) 和 (i_2,j_2) 之间的距离 $d(v_1,v_2)$ 为

$$d(v_1,v_2)=\begin{cases} i_1-i_2, & \text{如果 } i_1-i_2=j_1-j_2 \\ \infty, & \text{否则} \end{cases}$$

多重过滤是基于下面的观察。

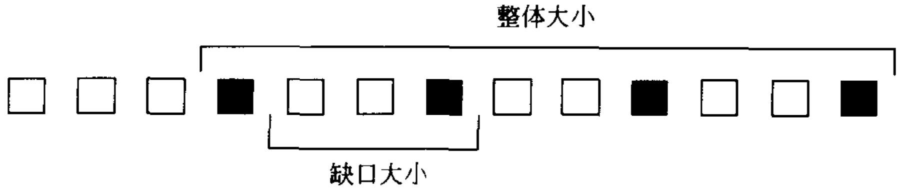


图 6.7 带缺口 4 元组

引理 6.6 设字符串 $x\cdots x_m$ 和 $y_1\cdots y_m$ 以至多 k 个误配而匹配，并且 $l=\left\lfloor\frac{m}{k+1}\right\rfloor$。那么这些字符串都有一段共同的连续 l 元组和一段缺口长为 $k+1$ 的带缺口 l 元组，它们之间的距离 d 满足条件 $-k\leqslant d\leqslant m-l$。

这个引理是双重过滤算法的基础，该算法用于带 k 个误配的查询匹配。

双重过滤算法 探测查询序列和文本序列之间的所有 m 个匹配直到有 k 个误配为止。

• 潜在匹配探测 找出所有查询和文本之间的连续 l 元组匹配，使之同缺口大小为 $k+1$ 的带缺口 l 元组匹配之间的距离在 $-k\leqslant d\leqslant m-l$ 范围内。

• 潜在匹配检验 通过向左向右延伸检验每一个潜在的匹配，直到或者第一次出现 $k+1$ 个误配，或者已到了查询序列或文本序列的两端。

引理 6.6 保证了双重过滤算法找出所有带 k 个误配的匹配。双重过滤算法的效率对一个大范围的参数来说比 l 元组过滤算法的效率大致要大 $\frac{A^{l-\delta}}{m\left(1-\frac{1}{A}\right)}$ 倍（这里，$\delta=\left\lceil\frac{l}{k+1}\right\rceil$）。

有一个错配的 9 元组的近似模式匹配可以简化成 4 元组的精确模式匹配。称之为一个从（9，1）模式匹配到（4，0）模式匹配的简化。能否把一个（9，1）模式匹配转变成一个（6，0）模式匹配呢？这样的话，可以提高过滤的效率。答案是肯定的，如果考虑如图 6.8 所示的带缺口的 6 元组。另一个问题是当从 (m,k) 模式匹配简化成 (m',k') 匹配时能否获得提速，其中 $0<k'<k$。对这个问题的探索导致了次线性模式匹配算法的发展[62,245]。

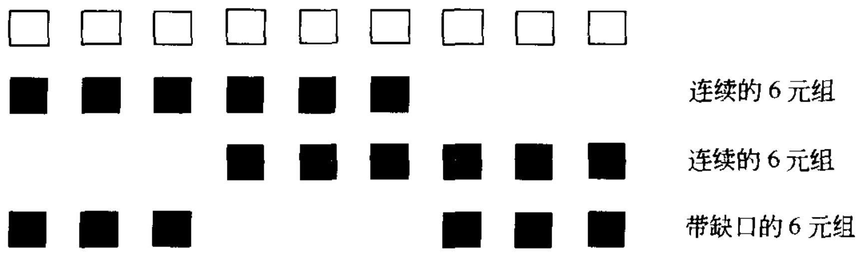


图 6.8　每个带一个错配的 9 匹配包含一个连续的 6 匹配或者一个带缺口的 6 匹配

6.15　若干其他的问题和方法

6.15.1　参数的序列联配

序列联配对插入/缺失以及替换罚分的选择是很敏感的，对这些参数的错误选择就可能导致生物学上不正确的联配。对生物学的应用来说，明白这些参数对联配结果的影响并选择适当的联配是非常重要的。在最简单的模型中，联配得分定义为＃个匹配$-\mu$＃个错配$-\sigma$＃个插缺，不同的（μ，σ）值对应于不同的最优联配。然而，在（μ，σ）空间中的某些区域却对应于相同的最优联配。实际上，（μ，σ）空间可以分解为凸多边形，使得同一多边形中的任何两点对应于相同的最优联配[107]。Waterman 等[360]、Gusfield 等[146]以及 Zimmer 和 Lengauer[379]都介绍过计算参数空间的多边形分解的高效率算法。

6.15.2　联配统计量和相变

Arratia 和 Waterman[13]研究了两条随机序列的局部联配得分的统计特性。结果是联配统计量严重依赖于插缺和错配罚分的选择。局部联配得分先是呈对数增长，然后随着序列的长度呈线性增长，并依赖于缺口罚分。增长的这两个区域被称为“指数区”和“线性区”，这两个区域之间的曲线被称为相变曲线。

设 $v_1\cdots v_n$ 和 $w_1\cdots w_n$ 为两条随机独立同分布的字符串。令 $S_n=S_n(\mu,\delta)$ 和 $H_n=H_n(\mu,\delta)$ 为对应于这两条字符串全局和局部联配得分（＃个匹配$-\mu$＃个错配$-\sigma$＃个插缺）的随机变量。

Arratia 和 Waterman[14]证明了

$$a=a(\mu,\delta)=\lim_{n\to\infty}\frac{S_n}{n}$$

在概率上是存在的。而且$\{a=0\}=\{(\mu,\delta):a(\mu,\delta)=0\}$是一段连续的相变曲线把$[0,\infty]^2$分成两部分：$\{a<0\}=\{(\mu,\delta):a(\mu,\delta)<0\}$和$\{a>0\}=\{(\mu,\delta):a(\mu,\delta)>0\}$。

对于$(\mu,\delta)\in\{a>0\}$的情形，Arratia 和 Waterman[14]证明了$\lim\limits_{n\to\infty}\frac{H_n(\mu,\delta)}{n}=a(\mu,\delta)$。对于$(\mu,\delta)\in\{a<0\}$的情形，他们引入了一个常数$b=b(\mu,\delta)$使得

$$\lim_{n\to\infty}P\left[(1-\varepsilon)b<\frac{H_n(\mu,\delta)}{\lg(n)}<(2+\varepsilon)b\right]=1$$

当$(\mu,\delta)\in\{a>0\}$时，计算$a(\mu,\delta)$是比较困难的。特别地，计算$a(0,0)$将解答 Steele-Arratia 猜想。要了解联配显著性的估计以及在“指数区”和“线性区”中的参数选择，请看参考文献[52，346]。要了解计算一个局部联配得分的概率的快速算法，请看参考文献[365]。

6.15.3 次优序列联配

最优联配有时候在生物学上并不是正确的结果，于是就设法去寻求能产生一个Δ-次优联配的集合的方法，这些次优解偏离最优解至多为Δ[355]。这个问题等价于寻找编辑图中的次优路径[63,250]。

6.15.4 串联重复片段的联配

大部分联配算法都只考虑插入、缺失和替换，然而也有其他的突变形式存在。串联重复片段就是一种突变事件，它是一段 DNA 序列被复制成一个或更多新的拷贝，这些拷贝都是一段紧接一段的形式。这种突变是相当普遍的，估计占了人类基因组的 10%。这些前后重复片段牵涉了一些人类的遗传疾病，其中有亨廷顿舞蹈症。Benson[30]提出了一个带串联重复片段的序列联配的有效算法。用于探测串联重复片段的算法是由 Landau 和 Schmidt[212]、Milosavljevic 和 Jurka[237]以及 Benson[31]提出的。

6.15.5 筛选数据库中的搜索结果

在数据库中搜索生物学上重要区域的匹配时经常会被其他一些匹配结果混杂。在序列某一区域的大量匹配可能会掩盖得分较低但却非常重要的其他地方的一些匹配。因为数据库搜索程序经常只给出一个固定数目的最高匹配结果并截去后面的输出，因此就需要设定规则使得揭示所有重要结果的匹配子集能显示出来。这个问题可由一列与联配得分相关的间隔（联配区域）来模拟。如果间隔I以一个很高的得分包含在间隔J中，那么I由J支配。筛选问题就是去鉴别和舍弃被一固定数目的间隔支配的那些间隔。Berman 等[34]用 BLAST 的一个版本在$O(n\lg n)$时间内解决了这个筛选问题，其中n是间隔数。

6.15.6 文本之间的统计距离

令$\mathcal{X}$为一字符串集合——比如，所有l元组的集合（Blaisdell，1988[36]）或

带缺口的 l 元组的集合[239]，对一个小的 l。给定一条字符串 $x\in \mathcal{X}$ 和一段文本 T，定义 $x(T)$ 为 x 在 T 中出现的次数（或频率）。Blaisdell[36] 及 Mironov 和 Alexandrov[239]定义了文本 V 和 W 之间的统计距离为

$$d(V,W)=\sqrt{\sum_{x\in \mathcal{X}}[x(V)-x(W)]^2}$$

这个定义远早于 Internet 搜索引擎用于寻找 Web 上的相似页面而开始使用的相关度量。用于寻找相似性的统计距离的效率在 Pevzner[266] 内有研究。统计距离方法在 BLAST 上的优势就在于速度：统计距离在数据库预处理时计算得非常快。缺点是统计距离会错过那些不保存共有的 l 元组的弱相似性。结果，这个算法的主要应用是在“数据库对数据库”的比较上，比如 EST 聚类方法。为了达到巨大的 EST 数据库所要求的很高速度，统计距离方法近来被应用于下标阵列中[55]。

6.15.7 RNA 折叠

RNA（核糖核酸）形成了一种复杂的三维结构，这对于信号识别和基因调控是非常重要的。RNA 中的一些有着 Watson-Crick 互补的碱基对可形成氢键。键（i，j）和（i'，j'）是交错的，如果 $i<i'<j<j'$，否则是非交错的。RNA 折叠问题的一个非常简单的表述就是试图找出非交错键的最大集。这个问题可以用动态规划算法来解决[253,354]。在一个更合适的模型中，人们试图去寻找一个 RNA 折叠的最小自由能[363,380,381]。然而，这些算法并不非常可信。一种更为有前景的方法是通过相关 RNA 分子的多序列联配来获得目标 RNA 的折叠。Eddy 和 Durbin[95]研究过一个考虑折叠信息的 RNA 多序列联配的问题。

7 多重联配

7.1 引言

蛋白质序列比较的目的是发现蛋白质间的“生物学的”（也就是结构的或功能的）相似性。生物学上相似的蛋白质可能并不表现出强的序列相似性，即使当序列间非常不同时，人们仍希望识别结构/功能的相似之处。如果序列间的相似性是弱的，那么成对联配不能识别生物学上相关的序列，因为弱的成对相似性对显著性不能成功地进行统计测试。许多序列同时进行比较通常能找到成对序列比较中不可见的相似性。引用 Hubbard 等[170]的话“成对联配轻声说……多重联配大声叫。”

简单的动态规划解决长度为 n 的 k 条序列的多重联配问题。由于这个方法的运行时间是 $O((2n)^k)$，许多不同变化、加速的基本算法已被提出来[298,299,364]。不过，精确多重联配算法对于大的 k 是不可行的[351]，于是，求次优多重联配的许多启发式方法也被提出来了。

一种自然的启发式方法是对 k 条字符串计算$\binom{k}{2}$次最优成对联配，然后将成对联配向最优联配靠拢，并以此方式将 k 条字符串结合在一起。不幸的是，并不是总能将成对联配结合为多重联配，由于某些成对联配是不相容的。因此许多多重联配算法尝试将一些相容的子集最优成对联配结合为多重联配。对于所有$\binom{k}{2}$个成对联配的一些小的子集这是可行的。这个问题要决定哪些成对联配的子集被选择用于处理。

最简单的方法是使用成对联配反复地增加一条字符串到增长着的多重联配。Feng 和 Doolittle[100]利用一对具有最大相似性的字符串，并将它们按“一旦有一个缺口就永远有这个缺口”的原理“合并”到一条新的字符串。其结果是，k 条序列的多重联配化为了 $k-1$ 条序列的多重联配。许多其他迭代多重联配算法使用类似的策略[22,27,162,336]。

尽管 Feng 和 Doolittle[100]的算法对于相近的序列运行得很好，但是这个方法没有“性能保证”。多重联配的第一个“性能保证”具有近似比$2-\frac{2}{k}$的近似算法是由 Gusfield[144]提出的。算法的思想基于兼容算法的概念和使用“一旦有一

个缺口就永远有这个缺口”的原理。

对于多重 k-路联配，Feng 和 Doolittle[100] 以及 Gusfield[144] 使用最优成对联配（2-路）作为积木。这个方法的自然延伸是使用最优 3-路（或 l-路）联配作为 k-路联配的积木。不过，由于不能明确如何定义兼容 l-路联配以及如何结合它们，这个方法面临一些组合问题。Bafna 等[18] 对于这个问题提出了具有近似率 $2-\frac{l}{k}$ 的算法。

生物学家常常用点阵的形式描述两条序列间的相似性。点阵是每个元素为 0 或 1 的矩阵，在位置（i，j）是 1 表示第一条序列的第 i 个位置和第二条序列的第 j 个位置具有相似性。相似性标准则由于使用氨基酸的物理参数间的相关系数的纯组合而不同（举例来说，开始于第一条序列的位置 i 和第二条序列的位置 j 的具有至多 k 个错配的长度为 m 的一个匹配）。然而，没有准则能完美地从偶然相似性（噪声）中区分出真实的相似性（生物学上相关的）。在生物学的应用中，噪声掩饰了真实的相似性，这个问题是决定如何从点阵中过滤噪声。

共享生物学上有关的相似性的几条序列的有效性将有助于从点阵中过滤噪声。当给定 k 条序列时，可以计算 $\binom{k}{2}$ 个成对点阵。如果所有 k 条序列共享一个区域的相似性。那么这个区域将在所有 $\binom{k}{2}$ 个点阵中可见到。同时，噪声不可能在所有 $\binom{k}{2}$ 个点阵中全都出现。实际问题是颠倒这个观察：给定 $\binom{k}{2}$ 个点阵，寻找所有或几乎所有的 k 条序列共享的相似性，并过滤噪声。Vingron 和 Argos[344] 设计了一个由二维点阵组装成一个 k 维点阵的算法。

7.2 计算多重联配的得分

令 $\mathcal{A}$ 为一有限字母表，a_1，…，a_k 为字母表 $\mathcal{A}$ 上的 k 条序列（字符串）。为方便计，假定每条字符串包含 n 个字符。令 $\mathcal{A}'$ 表示 $\mathcal{A}\cup\{-\}$，其中“—”表示间隙。序列 a_1，…，a_k 的一个联配可用一个 $k\times m$ 的矩阵 A 确定，其中 $m\geq n$。矩阵的每个元素是 $\mathcal{A}$ 的一个成员，每行 i 按顺序包含字符串 a_i 的字符，且有 $m-n$ 个散布的间隙。同样，假定多重联配矩阵的每个列至少包含 $\mathcal{A}$ 中的一个符号。多重联配的得分定义为列的得分之和，最优联配定义为具有最小得分的联配。

一个列的得分可以用许多不同方法定义。一个直接的方法是对具有大的字母变异的列指配高的得分。例如，在多重最短共同超序列问题中，一个列的得分就定义为这个列中取自 $\mathcal{A}$ 中的不同字母的数目。在多重最长共同超序列问题中，如果列中所有字母相同，一个列的得分定义为 -1，否则为 0。在生物学上更适

当的最小熵方法中，多重联配的得分定义为列的熵之和。一个列的熵定义为

$$-\sum_{x\in \mathcal{A}'} p_x \lg p_x$$

其中 p_x 是第 i 列中字母 $x\in \mathcal{A}'$的出现频率。列越可变，其熵就越高。一个完全保守的列（如在多重 LCS 问题）具有的最小熵是 0。

最小熵得分抓住了好的联配的生物学概念，但在实际中很难作有效的分析。下面介绍共有区距离（distance from consensus）和成对和（sum-of-pairs，SP）得分，它们是易于分析的。

• 共有区的距离　一个联配的共有区是在多重联配的每列中具有最多共同字母的字符串。共有区的距离得分定义为联配中与该列的共有字母不相同的所有字母的总数。

• 成对和（SP 得分）　对于多重联配 $A=(a_{ih})$，序列 a_i 和 a_j 的成对联配 A_{ij} 的得分

$$是\ s(A_{ij}) = \sum_{h=1}^{m} d(a_{ih}, a_{jh})$$

其中 d 指 $\mathcal{A}'$的元素间的距离。联配 A 的成对和得分(SP 得分)由 $\sum_{i,j} s(A_{ij})$ 给出。在此定义中,联配 A 的得分是 A 在所有序列对 a_i 和 a_j 的投影的得分之和。假定距离 d 的度量性质,对于 $\mathcal{A}'$中所有的 x,y,z,满足 $d(x,x)=0$ 和 $d(x,z)\leqslant d(x,y)+d(y,z)$。

7.3　装配成对联配

Feng 和 Doolittle[100] 利用具有最大相似性的字符串对，按“一旦有一个缺口，将永远有这个缺口”原理将它们“合并”成一条新的字符串。其结果，k 条序列的多重联配就简化为 $k-1$ 条序列的多重联配（它们中的一条相应于被合并的字符串）。在算法的开始步骤，选择最接近的字符串的动机是，相近的字符串提供有关联配的最可靠的信息。

给定序列 $a_1,\cdots,a_k$ 的一个联配 A 和某些序列子集的一个联配 A'，如果 A 联配序列的字母与 A'以相同的方式联配所得结果一致，则称 A 与 A'是相容的。Feng 和 Doolittle[100] 观察到，给定任何顶点用不同序列 a_i 标记的树，并对树的每条边给定成对联配，那么存在 k 条序列的一个多重联配，它与每个成对联配相容。特别地，这个结论对于 k 个顶点的星也成立，也就是一棵树具有中心顶点和 $k-1$ 片叶子。

引理 7.1　对于任意的星及对其边的任意指定的成对联配 $A_1,\cdots,A_{k-1}$，存在一个 k 条序列的联配 A，它与联配 $A_1,\cdots,A_{k-1}$ 的每一个相容。

给定星 G，定义星联配 A_G 为与这个星的边最优成对联配相容的联配。联配

A_G 在$\binom{k}{2}$个成对联配中优化 $k-1$ 次 SP 得分。如何为一个好的星联配的问题仍为未解决的问题，直到 Gusfield[144] 证明了如果适当地选择星 G，星联配以比率 $2-\frac{2}{k}$近似最优联配。

令 $G(V, E)$ 为一（无方向）图，令 $\gamma(i, j)$ 为顶点 $i\neq j\in V$ 间具有距离 $d(i, j)$的（固定）最短路径。对于一条边 $e\in E$，定义通信花费（communication cost）$c(e)$为在 G 中使用边 e 的最短路径 $\gamma(i, j)$ 的数目。例如，在具有 k 个顶点的星中每个边的通信花费是 $k-1$。定义图 G 的通信花费为 $c(G)=\sum_{e\in E}c(e)=\sum_{i\neq j\in V}d(i,j)$。在所有 k 顶点图中，k 个顶点的完全图 H_k 具有最小通信花费$c(G)=\frac{k(k-1)}{2}$。称 $b(G)=\frac{c(G)}{c(H_k)}=2\frac{c(G)}{k(k-1)}$为 G 的标准化通信花费。对于具有 k 个顶点的一个星，$b(G)=2-\frac{2}{k}$。Feng 和 Doolittle 在 1987 年[100]利用一棵树将成对联配结合为一个多重联配，且得知这棵树的标准化通信花费与最后所得的启发式算法的近似率有关。

定义 $C(G)=(c_{ij})$ 为一个 $k\times k$ 的矩阵，如果（i, j）是 G 中的一条边 e 那么 $c_{ij}=c(e)$，否则 $c_{ij}=0$。于是，联配 A 的加权成对和得分是

$$\sum_{i,j}c_{ij}\cdot s(A_{ij})$$

为了记号的方便，使用矩阵点积表示联配的得分。于是，令 $S(A)=(s(A_{ij}))$ 为序列对的得分矩阵，则加权成对和得分为 $C(G)\odot S(A)$。令 E 为除主对角线全由 0 组成外其余元素全为 1 的单位矩阵，则联配 A 的（未加权）成对和得分为 $E\odot S(A)$。

一个联配的成对得分继承距离矩阵的三角不等式性质。即，对于任何联配 A，对所有的 i，j，k，有 $s(A_{ij})\leqslant s(A_{ik})+s(A_{kj})$。此观察隐含了下面的引理。

引理 7.2 对 k 条序列的任何联配 A 和一个星 G，有 $E\odot S(A)\leqslant C(G)\odot S(A)$。

7.4 多重联配的近似算法

令 $\mathcal{G}$ 为k 个顶点图中的星的集合。集合 $\mathcal{G}$ 是平衡的，如果 $\sum_{G\in\mathcal{G}}C(G)=pE$，对于某个数量 $p>1$。例如，具有 k 个不同中心顶点的k 星是具有 $p=2(k-1)$ 的平衡集。由于 $C(G)$ 仅在星 G 的边上是非零的，因此星联配 A_G 成为 G 的边上的最优联配，对于任何联配 A，有 $C(G)\odot S(A_G)\leqslant C(G)\odot S(A)$。

引理 7.3 如果 $\mathcal{G}$ 是星的平衡集，那么

$$\min_{G\in\mathcal{G}} C(G)\odot S(A_G)\leqslant\frac{p}{|\mathcal{G}|}\min_A E\odot S(A)$$

证明 利用平均变量。

$$\min_{G\in\mathcal{G}} C(G)\odot S(A_G)\leqslant\frac{1}{|\mathcal{G}|}\sum_{G\in\mathcal{G}}C(G)\odot S(A_G)$$

$$\leqslant\frac{1}{|\mathcal{G}|}\sum_{G\in\mathcal{G}}C(G)\odot S(A)=\frac{p}{|\mathcal{G}|}E\odot S(A)$$

这里，不等式对任意联配 A 成立，特别对最优联配成立。 ■

引理 7.2 和引理 7.3 启发了 Align 算法。

(1) 构造一个星的平衡集 $\mathcal{G}$。

(2) 对于 $\mathcal{G}$ 中的每个星 G，装配为一个星联配 A_G。

(3) 选择一个星 G，使得 $C(G)\odot S(A_G)$ 在 $\mathcal{G}$ 中所有星上是最小的。

(4) 返回 A_G。

定理 7.1[144] 给定星的一个平衡集 $\mathcal{G}$，Align 算法返回一个在 $O(kn^2|\mathcal{G}|)$ 时间内具有 $2-2/k$ 性能保证的联配。

证明 注意到 $\frac{p}{|\mathcal{G}|}=(C(G)\odot E)/(E\odot E)=2-2/k$。Align 算法返回一个联配 A_G，它对于星 $G\in\mathcal{G}$ 是最优的，且得到最小得分 $\min_{G\in\mathcal{G}} C(G)\odot S(A_G)$。引理 7.2 和引理 7.3 意味着 $E\odot S(A_G)\leqslant C(G)\odot S(A_G)\leqslant(2-2/k)\cdot\min_A E\odot S(A)$。 ■

7.5 装配 l-路联配

在 k 个顶点上的一个 l-星 $G=(V,E)$ 由大小为 l 的 $r=\frac{k-1}{l-1}$ 个小集团定义，其顶点集仅在一个中心顶点相交（见图 7.1）。对于 $k=2t+1$ 个顶点的 3-星，$c(G)=(2t-1)2t+t$，且 $b(G)=2-\frac{3}{k}$。而对 $k=(l-1)t+1$ 个顶点的 l-星，则 $c(G)=[(l-1)t+1-l+1](l-1)t+t\left[\frac{l(l-1)}{2}-l+1\right]$，且 $b(G)=2-\frac{l}{k}$。在以 c 为中心的一个 l-星 G 中，一条边 e 的通信花费是

$$c(e)=\begin{cases}k-l+1, & \text{如果 } e \text{ 与 } c \text{ 相连}\\ 1, & \text{其他情形}\end{cases}$$

注意到一个 l-星 G 的通信花费矩阵

$$C(G)\odot E=(k-l+1)(k-1)+\frac{k-1}{l-1}\binom{l-1}{2}=\binom{k}{2}\left(2-\frac{l}{k}\right)$$

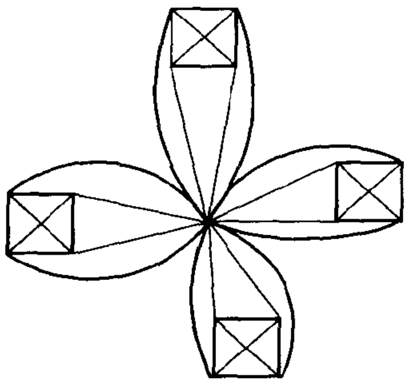


图 7.1 具有 4 个小集团的 l-星

令 A_1，…，A_r 为 l-星中的 r 个小集团的一个联配，每个 A_i 联配 l 条序列。于是有构造相似于 Feng 和 Doolittle[100] 的结果，且与引理 7.1 相类似的引理。

引理 7.4 对任何 l-星及对其小集团的任何指定联配 A_1，…，A_r，有 k 条序列的一个联配 A 与每个联配 A_1，…，A_r 相容。

可以归纳关于 l-星的 Align 算法，并证明与引理 7.2 和引理 7.3 类似的引理。作为结果，关于 l-星可归纳为定理 7.1，由此导致与 l-星的标准化通信花费 $2-\frac{l}{k}$ 相等的性能保证的一个算法。这个算法的运行时间是 $O(k(2n)^l|G|)$。因此，问题是寻找 l-星的一个小的平衡集 G。

已将多重联配问题简化为在平衡集 G 上对每个 l-星中的每个小集团寻找一个最优联配。寻找一个平衡集 G 有多困难？一个平凡的候选者简单地是所有 l-星的集合，由于对称它显然是平衡的。对于 $l=2$，Gusfield[144] 阐明了这样的事实，仅有 k 个 2-星就可构造一个具有近似比 $2-\frac{2}{k}$ 的多重联配算法。这实际上是个特殊的例子，对于 $l>2$，l-星的数目以指数 k 增长，使得基于所有 l-星生成的算法在计算上不可行。构建 l-星的一个小平衡集合并不是件无价值的事。Pevzner[265] 通过将问题映射为图中的最大化匹配，解决了 $l=3$ 的问题。Bafna 等[18]对任意 l 进一步设计了一个 $2-\frac{l}{k}$ 的近似算法。

7.6 点阵和图像重构

给定 k 条序列，容易产生 $\binom{k}{2}$ 个成对点阵。但要将这些成对点阵装配为 k 维点阵并寻找所有或几乎所有 k 条序列共有的相似性区域则不是一件容易的事。为了解决这个问题，Vihinen[343] 和 Roytberg[290] 提出了一个“叠加”成对点阵的设想，通过选择一条参考序列并将所有与它关联的其他序列叠加上去。下面介绍由二维点阵装配为 k 维点阵的[344]算法。

以简单的几何框架叙述装配成对联配的问题。在 k 维空间中考虑 M 个整数点

$$(i_1^1,\cdots,i_k^1),\cdots,(i_1^M,\cdots,i_k^M)$$

对于这些点，并不知道它们的坐标。假定观察这些点在每个维对 s 和 t，$1\leqslant s<t\leqslant k$ 上的投影

$$(i_s^1, i_t^1), \cdots, (i_s^M, i_t^M)$$

其他点（噪声）也一样。再假定不能区分代表真实投影的点与代表噪声的点。k 维图像重构问题是给定坐标 s 和 t 上的$\binom{k}{2}$个投影（含噪声），其中 $1 \leqslant s < t \leqslant k$，重构 M 个 k 维的点。

在这个构造下，k 条生物序列共有的每个相似性（共有元件）对应于 k 维空间中的一个整数点（i_1，…，i_k），其中 i_s 是第 s 条序列中的共有区的坐标，$1 \leqslant s \leqslant k$。在实际中，很难寻找到对应于共有区的整数点（$i_1$，…，$i_k$）。另一方面，容易寻找（尽管有相当大的噪声）到所有共有区（i_1，…，i_k）在每个坐标对 s 和 t 上的投影（i_s，i_t）。这个观察建立了多重联配问题与 k 维图像重构问题之间的连接。

从侧平面上给定的点，建议仅保留那些实现下述一致性准则的点：投影 s，t 的点（i，j）称为一致的，如果对于每个其他维数 u 存在一个整数 m，使得（i，m）属于投影 s，u，且（j，m）属于投影 t，u[135]。显然每个“真正的”点，即每个作为 k 维点的一个投影而产生的点是一致的。相反，代表噪声的随机点被认为是不一致的。此观察允许过滤掉大部分（虽然不可能全部）噪声，并导出 Vingron 和 Argos[344]的相乘和比较点阵算法。

7.7 点阵相乘的多重联配

将$\binom{k}{2}$个点阵的集合建为一个 k 分裂图 $G(V_1 \cup V_2 \cup \cdots \cup V_k, E)$，其中 V_i 是第 i 条序列的位置集。如果在比较序列 s 和 t 的点阵位置(i,j)存在一个点，将顶点 $i \in V_s$ 和顶点 $j \in V_t$ 通过一条(无方向)边 e 连接。边 $e \in E$ 将被记为 $e=(s,i|t,j)$ 表示它连接顶点 $i \in V_s$ 和顶点 $j \in V_t$。通过三条边$(s,i|t,j)$，$(t,j|u,m)$，和$(s,i|u,m)$形成的三角形表示为$(s,i|t,j|u,m)$。定义边$(s,i|t,j)$为一致的，如果对每个 $u \neq s,t$，$1 \leqslant u \leqslant k$，及某个 m 存在一个三角形$(s,i|t,j|u,m)$。子集 $E' \subseteq E$ 被称为一致的，如果对所有边$(s,i|t,j) \in E'$，存在三角形$(s,i|t,j|u,m)$，$\forall u \neq s$，t，$1 \leqslant u \leqslant k$，这些三角形的所有边在 E' 中。k-分裂图 G 定义为一致的，如果它的边集是一致的。显然，如果 $G'(V,E')$ 和 $G''(V,E'')$ 是一致图，那么它们的并 $G(V, E' \cup E'')$ 是一致的。因此，可以将任何 k-分裂图联合为一个惟一的最大一致性子图。感兴趣的是下面的问题。

图一致性问题 寻找一个 n-分裂图的最大一致性子图。

序列 s 和序列 t 的点阵被定义为邻接矩阵 A_{st}

$$(A_{st})_{ij} = \begin{cases} 1, \text{如果}(s,i|t,j) \in E \\ 0, \text{其他情形} \end{cases}$$

每个这样的矩阵对应于 E 的一个子集，将运算$\cup$，$\cap$和$\subset$应用于矩阵 A_{st}。再用邻接矩阵[344]的布尔乘法（用“$\circ$”表示）来形式地表示上面的一致性定义。如果 k-分裂图是一致的，当且仅当

$$A_{st} \subseteq A_{su} \circ A_{ut},\quad \forall s,\ t,\ u: 1 \leqslant s,\ t,\ u \leqslant k,\ s \neq t \neq u \tag{7.1}$$

特征化的式（7.1）建议用下面的简单过程解决一致性问题：仅保留邻接矩阵中的 1 的元素，这些元素将出现于矩阵本身及所有乘积 $A_{su} \circ A_{ut}$ 中。对于所有邻接矩阵执行这一过程，同时将改变用于乘积的矩阵。这导出了迭代矩阵相乘算法（上标区分不同迭代），从给定的 k-分裂图的邻接矩阵 $A_{st}^{(0)} := A_{st}$ 开始，定义

$$A_{st}^{(l+1)} := A_{st}^{(l)} \cap \Big(\bigcap_{u \neq s,t} A_{su}^{(l)} \circ A_{ut}^{(l)} \Big)$$

一旦对所有指标 s 和 t 完成计算，这个过程将被重复直到 m 次迭代，有 $A_{st}^{(l+1)} = A_{st}^{(l)}$，对于所有 $1 \leqslant s,\ t \leqslant k$ 成立。

点阵相乘算法[344]收敛于最大一致性子图，且每一迭代需要 $O(L^3k^3)$ 的时间（L 是序列的长度）。因为迭代的次数会是非常大的，Vingron 和 Pevzner[345]设计了整个运行时间为 $O(L^3k^3)$的算法，它与矩阵相乘算法的一次迭代的运行时间相当。在实际应用中，由于算法的输入数据是稀疏矩阵，这将使得算法更快。以所有$\binom{k}{2}$个点阵中的全部点的数目 M 表示，运行时间则为 $O(kLM)$。

7.8 若干其他的问题和方法

7.8.1 借助于进化树的多重联配

除了序列a_1，…，a_k 外，生物学家往往知道（或者假定他们知道）这些序列的进化历史（用进化树表示），这是经常会碰到。在这种情形下，序列 a_1，…，a_k 被指配为树的叶子，而问题就成为重构祖先序列（对应于树的内部顶点），它是树的边上全部突变数的最小化。在树中一个边的得分是分配到端点的序列间的编辑距离，进化树的得分则是整个树的所有边的边得分之和。对于给定进化树最优多重联配是分配序列到树的内部顶点而产生的最小得分[298]。Wang 等[352]和 Wang 和 Gusfield[350]为进化树联配发展了性能保证的近似算法。

7.8.2 在编辑图中切割角

Carrillo 和 Lipman[58]和 Lipman 等[224]提出了多重联配的分支边界技术。这个方法的思想是基于这样的观察，如果被一个多重联配使用的一个成对联配是不好的，那么整个多重联配将得不到好的得分。这个观察意味着好的多重联配使用好的成对联配，因此将搜索限制在 k 维联配矩阵的主对角线附近。

8 发现 DNA 中的信号

8.1 引言

HindⅡ限制酶发现以后，在 1970 年，Hamilton Smith 发现了也许是 DNA 中的第一个信号。限制酶的回文位点是最初的 DNA 切断信号。在 20 世纪 70 年代，寻找这个位点的序列并不是一个简单的问题：实际上，Hamilton Smith 连续发表了两篇关于 HindⅡ 的文章，一篇是关于酶的纯化，另一篇是关于发现酶的识别信号[196]。

回顾 20 世纪 70 年代早期，认为 Hamilton Smith 是幸运的：酶的限制位点是 DNA 中最简单的信号。30 年后，它们可能仍然是 DNA 中可以找到的仅有的信号。大多数其他信号（启动子，剪接位点等）是如此复杂，以至于仍然没有好的模型和可靠的算法去识别它们。

计算生物学中的主要挑战是了解基因调控。例如，基因表达的调控可能包括蛋白质结合到 DNA 的一个区域以影响相邻基因的转录。因为蛋白质与 DNA 的结合机制仍然没有充分的了解以至可以用计算机来预测结合位点的位置，而常规的实验方法则是确定结合位点的大概位置。这些实验通常导致长度为 n 的 DNA 片段的鉴定，其中包含长度 $l \ll n$ 的结合位点（一个未知的有魔力的词）。当然，这样的实验是不足以用来寻找结合位点的，但是实验性地发现 DNA 片段的样品给了研究人员发现魔力词（magic word）的希望。

作为最简单的形式，信号发现问题（特别是限制性酶切位点问题）可以用下面的方法来阐明。假设有 K 条序列样品，再假设有一个（未知的）魔力词出现在这些序列的不同（不知道的）位置。能够找到这些魔力词吗？

求解魔力词问题的一个普通的方法是，测试所有长度为 l 的词，并且寻找那些出现在所有（几乎所有）样品序列中的魔力词[326,338,370]。如果魔力词是在样品中惟一频繁地出现的词，那么问题就（可能）解决了。否则，将增加长度 l，并重复以上过程。

对于那些较短的词，例如 GAATTC，EcoRⅠ的限制位点，上述方法将会很有效。但是，如果样品中的序列长度接近 4^6，那么随机词就会和魔力词竞争，因为一些随机词有可能很偶然地出现在许多序列中。如果样品中的核苷酸频率呈偏斜分布，情况将会变得更加复杂。

当魔力词中间有间隙时，如 $CCAN_9TGG$ 及 XcmⅠ限制酶切位点（N 代表任何核苷酸，而 N_9 则表示位点中长度为 9 的间隙），问题将会变得更复杂。当然，可以列举所有的有此间隙的模式，但是问题的计算复杂性将会飞速增长，特别是如果允许模式有多个间隙时。甚至例如寻找相当简单的魔力词“字母-间隙-字母-间隙-字母”，问题也变得不再简单了。在发现了第一个限制位点模式[318]的 20 年后，由 Hamilton Smith 和他的同事们发表了其他的模式寻找方法的文章。另外一个使人畏缩的识别任务是寻找像 $Pu^mCN_{40-2000}Pu^mC$（*Pu* 代表 A 或 G），McrBC 内切核酸酶的识别位点的信号。

上述问题都不是很简单的，而真正的关于信号寻找的生物问题更为复杂。困难之处在于生物信号可能更长、有间隙并且模糊。例如，大肠杆菌（*E. coli*）启动子的魔力词是 $TTGACAN_{17}TATAAT$。由于计算的复杂性，列举和检验所有的有此类型的模式几乎是不可能的。而且，即使列举了所有的有此类型的模式也没什么用，因为上面的模式代表了一种理想的启动子，它并没有在已知的启动子序列中出现过。更确切地，它是所有已知启动子的共有序列：既不是共有序列碱基，也不是两部分信号之间的间隙。换句话说，在这个例子中的魔力词比较像“具有一个间隙及最大 4 个错配的 12 个非退化的位子”。仍然没有发现这种类型信号的可靠的方法。现有算法的缺点是，对那些微小的信号，这些方法往往收敛到局部最小，而局部最小更有可能代表了随机模式，而不是信号。

8.2　Edgar Allan Poe 和 DNA 语言学

当 William Legrand 从 Edgar Allan Poe 的小说“金甲虫”中发现了一张由海盗船长 Kidd 写的羊皮纸。

```
53‡‡†305))6*;4826)4‡)4‡):806*;48†8¶60))8
5;1‡(;:‡*8†83(88)5*†;46(88*96*?;8)*‡(;485
);5*†2:*‡(;4956*2(5*--4)8¶8*;4069285);)6†8
)4‡‡;1(‡9;48081;8:8‡1;48†85;4)485†528806
*81(‡9;48;(88;4(‡?34;48)4‡;161;:188;‡?;
```

他的朋友告诉他：“是否 Golconda 的所有的宝石都在等待我解决这个迷呢？但是我知道我不可能解决它。”Legrand 先生回答他：“我们怀疑是否人类的智慧可以建立一个谜，而这个谜仅依靠人类的智慧又是不可能解决的。”他注意到有 3 个符号——；4 8 的组合在这段文字中频繁出现。他也知道 Kidd 船长的海盗们是说英语的，而最频繁的英语单词是“the”。假定；4 8 编码“the”，Legrand 先生译解了羊皮纸，并找到了海盗们的宝石（和一些骨骼）。有此见识后，Legrand 先生有一个稍微简单一些的文本要译解。

```
53‡‡†305))6*THE26)H‡.)H‡):E06*THE†E¶60))E
5TI‡(T:‡*E†E3(EE)5*†T46(EE*96*?TE)*‡(THE5
)T5*†2:*‡(TH956*2(5*--H)E¶E*TH0692E5)T)6†E
)H‡‡TI(‡9THE0E1TE:E‡1THE†E5TH)HE5†52EE06
*E1(‡9THET(EETH(‡?3HTHE)H‡T161T:1EET‡?T
```

你可以试着去计算出“（”的编码，并且完成这个解密工作。

DNA 文本并不是容易译解的，但是大自然是否可以构想出一种以人类的智巧也无法破解的谜，对此仍有一些疑问。然而，DNA 语言学借鉴了 Legrand 先生的科学方法，DNA 语言学中的一个流行的方法就是假设频繁出现的或很少出现的词对应于 DNA 中的一个信号。如果一个词比预期出现的次数多（或少），那它就是一个可能的“信号”，随之产生了关于这个词的“生物学”意义的问题[47,53]。例如，Gelfand 和 Koonin[124]证明了在 archaeon *M. jannaschii* 中的最可能消除的 6-回文序列很有可能是限制-调整系统中的识别位点。

DNA 语言学是用于信号寻找的模式驱动方法的核心，这是基于列举所有的可能的模式，并且在其中寻找最频繁的或最适合的模式[46]。适应性测量已知信号的统计显著性估计与近似匹配这一信号的片段的信息量之间的变化。模式驱动方法包括以下步骤。

- 定义适应性度量（例如频率）。
- 计算每一个词与有关的 DNA 片段样品的适应度。
- 报告作为可能信号的最合适的词。

模式驱动方法的问题是效率问题，因为长度为 l 的模式的搜索空间是 $|\mathcal{A}|^l$，其中 $\mathcal{A}$ 是字母表。为了减少搜索，可以使用基于 Karp-Miller-Rosenberg 算法的思想[188]，它是基于这样的观察，如果一段字符串出现在 k 个序列中，那么它的所有的子字符串至少也出现在 k 个序列中。因此，每一个频繁的字符串可以由频繁的子字符串装配得到。例如，一个简单的方法是依据包含所有频繁的 l 元组的清单创建一个所有频繁的 $2l$ 元组的清单，这可由连接每个成对的频繁的 l 元组，随后检查所有这些连接的频率来实现。关于这个问题的另一个解决的方法是利用后缀树[145]。

为了在文本中寻找频繁出现的或很少出现的词，人们不得不计算每一个词 W 出现次数（频率）的期望值 $E(W)$ 和方差 $\sigma^2(W)$。然后，频繁出现的或很少出现的词被看成是偏离预期频率的词。在许多 DNA 语言学论文中，一个词在文本中的出现次数的方差 $\sigma^2(W)$ 错误地被假设成了期望值 $E(W)$。

查明一个词在文本中 k 次出现的概率问题涉及到生成函数和复杂分析[141]。困难之处是一个词在文本中出现的概率不仅依赖于这个词的长度，还依赖于这个词的重叠部分的结构，这由自相关多项式决定[141]。例如，AAA 出现次数的分

布（自相关多项式 $1+x+x^2$）与 ATA 出现次数的分布（自相关多项式 $1+x^2$）完全不同，甚至在有等概率字母表的随机 Bernoulli 文本中（重叠的词矛盾）也是不同的。下面，将讨论对一个傻子解释重叠的词矛盾这一意思的最好方法。

8.3 适合傻子的最好的赌博

John Conway 研究出重叠的词矛盾是适合傻子的最好的赌博的基础。适合傻子的最好的赌博由两位选手开始，这两位选手从 0-1 字母表中选择长度为 l 的词。第一个选手掷铜板选择了长度为 l 的序列 A，第二个选手知道了 A 是什么，于是就选择了另一条长度为 l 的序列 B。然后他们扔硬币，直到 A 或 B 作为一组 l 长度的连续结果出现。这个比赛将会以概率 1 结束。

在 Elkhorn 的一天晚上，Shorty 对 Smoke 说："怪了，你怎么不赌？难道这就是你的本性吗？"（引自 Jack London 的小说"Smoke Bellew"）。Smoke 回答说："是的，但是统计数字在我的头脑中，对于我的钱，我喜欢条件均等的机会。"

初看起来好像 A 和 B 对于他们的钱有一个条件均等的机会。尽管有些人意识到，在这个比赛中一些词比另一些词要多，看起来好像在选择"最多的"词后，A 应该赢。这个游戏激起人们兴趣的特性是，如果 $l\geqslant 3$，不管 A 是什么，第 2 个选手能够选择 B 打败 A。更奇特的是，适合傻子的最好的赌博是一个不传递的比赛：A 打败 B，B 打败 C，不意味着 A 打败 C（还记得石头，剪刀，布吗?）。

假设 A 选择 00，B 选择 10。在两次投掷后，不是 A 赢（00）就是 B 赢（10），或者游戏继续（01 或 11）。然后，因为无论如何 B 总会赢，所以 A 继续比赛就没有意义了。因此在这个比赛中成败的可能性 B 比 A 是 3∶1。

适合傻子的最好的赌博的分析基于相关多项式[141]。给定两个 l-字母的词的 A 和 B，A 和 B 的相关性由 $AB=(c_0, \cdots, c_{l-1})$ 表示，这是一个 l-字母的布尔词（见图 8.1）。如果 B 的 $(n-i)$ 前缀（最开始的 $n-i$ 个字母）与 A 的 $(n-i)$ 后缀（最后的 $n-i$ 个字母）一致，则 AB 的第 i 位定义为 1。否则，AB 的第 i 位定义为 0。A 和 B 的相关多项式定义为 $K_{AB}(t)=c_0+c_1t^1+\cdots+c_{l-1}t^{l-1}$。又记 $K_{AB}=K_{AB}\left(\frac{1}{2}\right)$。

John Conway 提出了计算 B 会赢 A 的概率的最好的公式

$$\frac{K_{AA}-K_{AB}}{K_{BB}-K_{BA}}$$

Conway 的这个公式的证明从来没有公布过。Martin Gardner[118] 关于这个公式写道：我不知道它是怎么得到的。它好像是通过魔力计算得到的结果，就像

Conway 的其他许多算法一样。

Li[222] 利用数学技巧独立地给出了这个公式的证明，Guibas 和 Odlyzko[141] 利用生成函数也独立地给出了这个公式的证明。在下面一节中，将给出 Conway 等式的一个简短的证明[268]。

8.4　Conway 等式

令 $AB=(c_0,\cdots,c_{l-1})$ 表示 A 和 B 的相关性，又令 c_{m_1}，…，c_{m_k} 表示 AB 等于 1 的单元。长度为 m_1，…，m_k 的词 $A=a_1\cdots a_l$ 的 k 个前缀的集合表示为 $\mathcal{H}_{AB}$（见图 8.1）。

$$(a_1\cdots a_{m_1}),(a_1\cdots a_{m_1}\cdots a_{m_2}),\cdots,(a_1\cdots a_{m_1}\cdots a_{m_2}\cdots a_{m_k})$$

		AB
$A=XYYXYY$		
$B=YYXYYX$	移位=0	0
$YYXYYX$	移位=1	1
$YYXYYX$	移位=2	0
$YYXYYX$	移位=3	0
$YYXYYX$	移位=4	1
$YYXYYX$	移位=5	1

$K_{AB}=t^1+t^4+t^5$

$H_{AB}=\{X,\ XYYX,\ XYYXY\}$

$K_{AB}\ (1/2)\ =1/2+1/16+1/32=P(H_{AB})$

图 8.1　词 A 和 B 的相关多项式 $[AB=(010011)]$ 和集合 $\mathcal{H}_{AB}$

给定两个词 X 和 Y，则 X 和 Y 的串联表示为 $X*Y$。给定两个词的集合 $\mathcal{X}=\{X\}$ 和 $\mathcal{Y}=\{Y\}$，则 $\mathcal{X}*\mathcal{Y}$ 表示所有的 $\mathcal{X}$ 中的词和 $\mathcal{Y}$ 中的词串联成的词 $\{X*Y\}$的集合。集合 $\mathcal{X}*\mathcal{Y}$ 包含 $|\mathcal{X}|\cdot|\mathcal{Y}|$ 个词（可能有重复）。

由 l 个布尔字母组成的词的概率记为 $P(X)=\frac{1}{2^l}$，它代表 l 次正面或反面试验的结果。对词的集合 $\mathcal{X}=\{X\}$，定义 $P(\mathcal{X})=\sum_{x\in\mathcal{X}}P(X)$，将采用下面的简单观察结果。

引理 8.1　$K_{AB}\left(\frac{1}{2}\right)=P(\mathcal{H}_{AB})$。

一个词 W 是 A-胜利的，如果它最后包含 A 而没有包含 B。如果 $W*A$ 是 A-胜利的，则词 W 是 A-预胜利的。定义 $\mathcal{S}_A$ 为所有的 A-预胜利的集合。可类似定义 B-胜利，B-预胜利和所有的 B-预胜利的集合 $\mathcal{S}_B$。

证明的思想是考虑所有的非-胜利的词。

$\mathcal{T}=\{T$：T 既不是 A-胜利，也不是 B-胜利$\}$。

每个词 $T*A$，其中 $T\in\mathcal{T}$ 对应于 A-胜利或 B-胜利。如果 $T*A$ 对应于 A-

胜利，那么 T 可以表示成 A-预胜利 $*$ H_{AA} 形式，其中 $H_{AA}\in\mathcal{H}_{AA}$［见图 8.2 (a)］。如果 $T*A$ 对应于 B-胜利，那么 T 可以表示成 B-预胜利 $*$ H_{BA}，其中 $H_{BA}\in\mathcal{H}_{BA}$［见图 8.2（b)］。这暗示着下面的非-胜利表示。

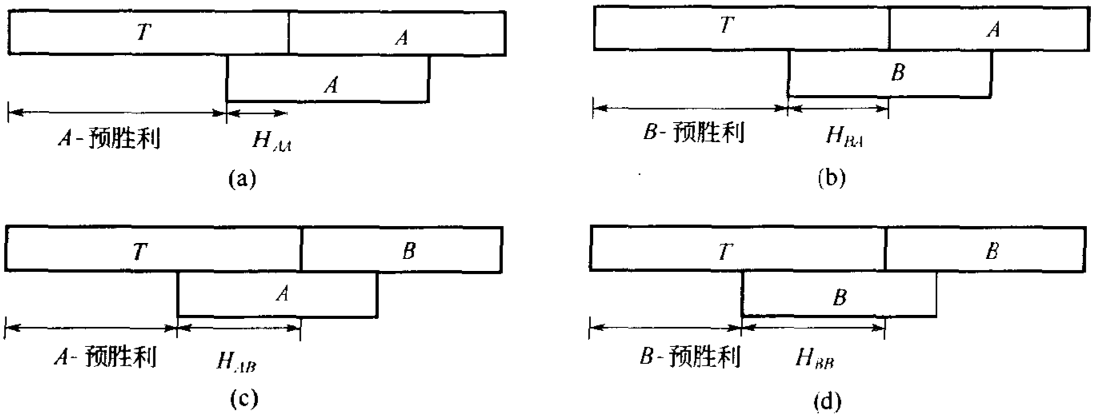


图 8.2 集合 $\mathcal{T}*\{A\}$ 和 $\mathcal{T}*\{B\}$ 中的词的不同表示

引理 8.2 $\mathcal{T}=\mathcal{T}_1=(\mathcal{S}_B*\mathcal{H}_{BA})\cup(\mathcal{S}_A*\mathcal{H}_{AA})$

类似地，每一个词 $T*B$，其中 $T\in\mathcal{T}$ 对应于 A-胜利或 B-胜利。如果 $T*B$ 对应于 A-胜利，那么 T 可以表示成 A-预胜利 $*$ H_{AB}，其中 $H_{AB}\in\mathcal{H}_{AB}$［见图 8.2（c)］。如果 $T*B$ 对应于 B-胜利，那么 T 可以表示成 B-预胜利 $*$ H_{BB}，其中 $H_{BB}\in\mathcal{H}_{BB}$［见图 8.2（d)］。这暗示了另一非-胜利的表示。

引理 8.3 $\mathcal{T}=\mathcal{T}_2=(\mathcal{S}_A*\mathcal{H}_{AB})\cup(\mathcal{S}_B*\mathcal{H}_{BB})$

定理 8.1 B 战胜 A 的概率是 $\dfrac{K_{AA}-K_{AB}}{K_{BB}-K_{BA}}$

证明 引理 8.1 和引理 8.3 表示在集合 $\mathcal{T}_2$ 中的词的所有概率是

$$\begin{aligned}P(\mathcal{T}_2)&=P(\mathcal{S}_A*\mathcal{H}_{AB})+P(\mathcal{S}_B*\mathcal{H}_{BB})\\&=P(\mathcal{S}_A)P(\mathcal{H}_{AB})+P(\mathcal{S}_B)P(\mathcal{H}_{BB})\\&=P(\mathcal{S}_A)K_{AB}+P(\mathcal{S}_B)K_{BB}\end{aligned}$$

类似地，引理 8.1 和引理 8.2 表示

$$P(\mathcal{T}_1)=P(\mathcal{S}_B)K_{BA}+P(\mathcal{S}_A)K_{AA}$$

根据引理 8.2 和引理 8.3，$\mathcal{T}_1$、$\mathcal{T}_2$ 代表同样的集合 $\mathcal{T}$；因此，$P(\mathcal{T}_1)=P(\mathcal{T}_2)$，并且

$$P(\mathcal{S}_A)K_{AB}+P(\mathcal{S}_B)K_{BB}=P(\mathcal{S}_B)K_{BA}+P(\mathcal{S}_A)K_{AA}$$

于是有

$$\frac{P(\mathcal{S}_B)}{P(\mathcal{S}_A)}=\frac{K_{AA}-K_{AB}}{K_{BB}-K_{BA}}$$ ■

8.5 DNA 中的频繁词

Gentleman 和 Mullin[125] 以及 Pevzner 等[269] 给出了在伯努利（Bernoulli）文

本中的一个词 W 的出现次数的方差 $Var(W)$ 的公式。在具有 l 个等概率字母的字母表中，对长度为 n 的 Bernoulli 文本来说

$$Var(W)=\frac{n}{l^k}\left[2K_{WW}\left(\frac{1}{l}\right)-1-\frac{2k-1}{l^k}\right]$$

其中 k 是词 W 的长度，$K_{WW}(t)$ 是词 W 和 W 的相关多项式（W 的自相关多项式）。

为了得到这个公式，在 l 个字母的字母表中考虑（为了简单起见）一个长度为 n 的循环文本，每一个字母每一个位置上的概率是 $\frac{1}{l}$。对一个固定的 k 个字母的词 W，如果 W 在文本的第 i 位开始，则定义随机变量 x_i 为 1，否则为 0。x_i 的平均值表示为 $p=\frac{1}{l^k}$。文本中 W 出现的次数由随机变量给出

$$X=\sum_{i=1}^{n}x_i$$

而平均值

$$E(X)=\sum_{i=1}^{n}E(x_i)=np$$

方差

$$Var(X)=E(X^2)-E(X)^2=\sum_{\{1\leqslant i,j\leqslant n\}}E(x_ix_j)-E(x_i)E(x_j)$$

在循环文本中，记 $d(i,j)$ 是位置 i 和 j 间的(最短)距离，那么

$$\begin{aligned}Var(X)=&\sum_{\{(i,j):d(i,j)\geqslant k\}}E(x_ix_j)-E(x_i)E(x_j)+\\&\sum_{\{(i,j):d(i,j)=0\}}E(x_ix_j)-E(x_i)E(x_j)+\\&\sum_{\{(i,j):0<d(i,j)<k\}}E(x_ix_j)-E(x_i)E(x_j)\end{aligned}$$

对于 $d(i,j)\geqslant k$ 而言，因为随机变量 x_i、x_j 是独立的，所以上述公式的第一项为 0，第二项是 $n(p-p^2)$，而第三项则可重写成下面的形式

$$\begin{aligned}&\sum_{\{(i,j):0<d(i,j)<k\}}E(x_ix_j)-E(x_i)E(x_j)=\\&\sum_{i=1}^{n}\sum_{t=1}^{k-1}\sum_{\{j:d(i,j)=t\}}E(x_ix_j)-E(x_i)E(x_j)\end{aligned}$$

固定 t，如果第 t 位的 WW 的相关系数是 1，则 $E(x_ix_{i+t})$ 等于 $p\,\frac{1}{l^t}$，否则等于 0。于是 $E(x_ix_{i+t})=c_tp\,\frac{1}{l^t}$，因为对每一个 i，有两个位置上的 j，使得 $d(i,j)=t$，

$$\sum_{t=1}^{k-1}\sum_{\{j:d(i,j)=t\}}E(x_ix_j)=2p\sum_{t=1}^{k-1}c_t\,\frac{1}{l^t}=2p\left[K_{WW}\left(\frac{1}{l}\right)-1\right]$$

因此

$$\sum_{\{(i,j):0<d(i,j)<k\}} E(x_i x_j) - E(x_i)E(x_j) = \sum_{i=1}^{n} \left\{ 2p\left[K_{WW}\left(\frac{1}{l}\right) - 1 \right] - 2(k-1)p^2 \right\}$$

$$= np\left[2K_{WW}\left(\frac{1}{l}\right) - 2 - 2(k-1)p \right]$$

并且

$$Var(X) = np\left[2K_{WW}\left(\frac{1}{l}\right) - 1 - (2k-1)p \right]$$

这个结果表明，词之间出现显著变化的频率的方差即使对 Bernoulli 文本也同样成立。特别是对于字母 A，T，G 和 C 等概率的四个字母的字母表来说

$$\frac{Var(\mathrm{AA})}{Var(\mathrm{AT})} = \frac{21}{13} \quad \text{以及} \quad \frac{Var(\mathrm{AAA})}{Var(\mathrm{ATG})} = \frac{99}{59}$$

因此在分析频繁词时，如果忽略重叠词的矛盾将会对统计显著性的估计造成 200%的误差。对两字母的字母表 Pur/Pyr 或 S/W 来说，忽略重叠词的矛盾将会对统计显著性的估计造成 500%(!)的误差$\left(\frac{Var(\mathrm{SS})}{Var(\mathrm{SW})} = \frac{5}{1}\right)$。

上面的公式使我们可以对 Bernoulli 文本计算方差。Fousler 和 Karlin[112]，Stuckle 等[330]以及 Kleffe 和 Borodovsky[199]给出了近似的公式，使得人们可以计算由马尔可夫（Markov）链产生的文本的方差。Prum 等[281]获得了马尔可夫链模型中的词的出现次数的正态极限分布规律。最后，Regnier 和 Szpankowski[282]研究了近似词的出现，得到了平均数、方差以及近似出现概率的精确的和渐近的公式。

8.6 共有词的分析

对一个词 W 和一个样品 $\mathcal{S}$，记 $W(\mathcal{S})$为从 $\mathcal{S}$ 中获得含有 W 的序列的数目。如果一个魔力词在样品中精确地出现，那么对每个由 l 个字母组成的词 W 来说，有个简单的计算 $W(\mathcal{S})$的方法，能够检测出魔力词为最频繁的词。当魔力词中允许有 k 个误差（错配）时，问题就变得更加复杂。对于这种情况，Waterman 等[358]和 Galas 等[115]提出了共有词（consensus word）的分析，它本质上是一个近似词的计算。对于每一个词 W，他们定义了一个邻域（neighborhood)，它包含与 W 的距离在 k 之内的所有的词，并且计算了样品中在这些邻域中的所有的词的出现情况。他们也引进了“加权出现（weighted occurrence)”的思想，对那些误差较少的邻域分配较高的权重。采用共有词的分析方法，Galas 等[115]能够检测出大肠杆菌启动子信号中的 TTGACA 和 TATAAT 共有序列。

用 $D_H(s,t)$ 表示相同长度的字符串 s 和 t 之间的 Hamming（汉明）距离。Mirkin 和 Roberts[238]指出，在某种意义下，近似词的计算等同于下述问题。

共有字符串问题 给定一个序列样品 $\mathcal{S}=\{s_1, \cdots, s_n\}$ 和整数 l，从每一个 s_i 中寻找长度为 l 的中间字符串 s 和长度为 l 的子字符串 t_i，使 $\sum_{i=1}^{n} d_H(s,t_i)$ 为极小。

Li 等[221]指出共有字符串问题是 NP 难题，并给出了求解这个问题的多项式时间近似计算方案（polynomial time approximation scheme，PTAS）。此算法是基于强字符串的概念。给定 n 个长度为 l 的字符串的集族 $t_1, \cdots, t_n$，若字符串 s 的第 i 位字母与 $t_1, \cdots, t_n$ 中的 n 个第 i 位字母相比是最频繁的，则该字符串就是对 $t_1, \cdots, t_n$ 而言的强字符串。Li 等[221]为共有字符串问题设计了 PTAS 算法，该算法基于对每 r 个长度为 l 的 $\{s_1, \cdots, s_n\}$ 的子字符串 $t_{i_1}, \cdots, t_{i_r}$ 选取强字符串。

将样品 $\mathcal{S}$ 中的多条序列连接成一条单一的复合序列常常是方便的，这就把寻找共有字符串问题转化为寻找文本中的最频繁字符串问题。这个问题的一个较天真的解法是寻找文本 T 中的每 l 个字母的字符串 W 的出现次数 $W(T)$。Apostolico 和 Preparata[10]为下述问题设计了一个有效的算法。

字符串统计问题 给定一个文本 T 和一个整数 l，对每个 l 个字母的字符串 W 寻找 $W(T)$。

如果考虑近似字符串的出现，那么字符串统计问题就变得很困难。设 $W_k(T)$ 是 W 在 T 中的近似出现次数，允许至多有 k 个误配。没有一个有效的方法来解决下面的问题。

频繁字符串问题 给定文本 T，整数 l 和 k，寻找 l 个字母的字符串 W 使 $W_k(T)$ 对所有 l 个字母的词为最大。

共有词分析是序列驱动方法用于信号发现的一个例子。序列驱动方法并不列举所有的模式（pattern），而仅仅考虑在样品中出现的模式。给定样品 $\mathcal{S}_1$ 中的频繁词集族 $\mathcal{W}_1$ 和样品 $\mathcal{S}_2$ 中的频繁词集族 $\mathcal{W}_2$，通过交叉 $\mathcal{W}_1$ 和 $\mathcal{W}_2$ 可得到 $\mathcal{S}_1 \cup \mathcal{S}_2$ 中的频繁词集族。给定 n 条序列的样品，可以将它看成 n 个样品的一个集合，并且开始组合频繁词集合，直到组合了所有的序列。不同的序列驱动方法在集合组合时的选取方法和组合集合的方法是不同的。

8.7 CG 岛和“公平赌场”

在许多基因组中最稀少的二核苷酸是 CG。因为 CG 中的 C 是典型的甲基化的，而甲基-C 有可能突变成 T。但是甲基化作用被一个区域周围的基因抑制着，这个区域就叫作 CG 岛，也就是 CG 出现相对频繁的区域。问题是怎样在基因组

中定义和找到CG岛。

从下面的“赌博”问题，可知找到CG岛并不是很难[93]。“公平赌场”中的一个商人可能使用一枚公平的硬币，也可能使用一枚动过手脚的硬币，使得这枚硬币有$\frac{3}{4}$的概率掷到正面。为了安全起见，商人不打算更换硬币——这很少发生，概率为0.1。给定硬币投掷序列，找出什么时候商人用了作假的硬币，而什么时候商人又用了公平的硬币。

首先，在假设商人从来不更换硬币的条件下来求解这个问题。问题是商人用的是公平的硬币$\left(p^+(0)=p^+(1)=\frac{1}{2}\right)$还是作假的硬币$\left(p^-(0)=\frac{1}{4},p^-(1)=\frac{3}{4}\right)$。如果投掷的结果序列是$x=x_1\cdots x_n$，那么公平的硬币产生的概率是$P(x|\text{公平硬币})=\prod_{i=1}^{n}p^+(x_i)=\frac{1}{2^n}$，而作假的硬币产生的概率是$P(x|\text{作假硬币})=\prod_{i=1}^{n}p^-(x_i)=\frac{1}{4^{n-k}}\frac{3^k}{4^k}=\frac{3^k}{4^n}$，其中$k$是$x$中1出现的次数。结果，当$k<\frac{n}{\log_2 3}$时，商人很可能使用了公平硬币；而当$k>\frac{n}{\log_2 3}$时，商人很可能使用了作假硬币。可以定义对数差异比率（log-odds ratio）如下。

$$\log_2\frac{P(x|\text{公平硬币})}{P(x|\text{作假硬币})}=\sum_{i=1}^{k}\log_2\frac{p^+(x_i)}{p^-(x_i)}=n-k\log_2 3$$

寻找CG岛的一个简单的方法就是对每一个固定长度的滑动窗口计算log-odds比率。得分为正数的窗口就可能是CG岛。这个方法的缺点是不能预先知道CG岛的长度。隐马氏模型提供了一个求解这个问题的不同的概率方法[69]。

8.8 隐马氏模型

隐马氏模型（Hidden Markov Model，HMM）$\mathcal{M}$是由一个字母表Σ、（隐含）状态Q的集合、状态转移概率矩阵A和发出概率矩阵E来定义的。

- Σ是符号的字母表。
- Q是从字母表Σ中发出符号的状态的集合。
- $A=(a_{kl})$是状态转移概率矩阵$|Q|\times|Q|$。
- $E=(e_k(b))$是发出概率矩阵$|Q|\times|\Sigma|$。

在“公平赌场”投掷硬币相当于下述的HMM。

- $\Sigma=\{0,1\}$，相当于反面（0）或正面（1）。
- $Q=\{F,B\}$，相当于公平的或作假的硬币。
- $a_{FF}=a_{BB}=0.9$，$a_{FB}=a_{BF}=0.1$。
- $e_F(0)=\frac{1}{2},e_F(1)=\frac{1}{2},e_B(0)=\frac{1}{4},e_B(1)=\frac{3}{4}$。

在隐马氏模型 $\mathcal{M}$ 中，路径 $\pi=\pi_1\cdots\pi_n$ 是状态的序列。例如，一个商人在最初三次和最后三次投掷中使用公平的硬币，而在中间五次使用作假的硬币，那么相应的路径为 $FFFBBBBBFFF$。由路径 π（给定模型 $\mathcal{M}$）而产生的序列 x 的概率是

$$P(x|\pi)=\prod_{i=1}^{n}P(x_i\mid\pi_i)P(\pi_i\mid\pi_{i+1})=a_{\pi_0,\pi_1}\cdot\prod_{i=1}^{n}e_{\pi_i}(x_i)\cdot a_{\pi_i,\pi_{i+1}}$$

为了方便起见，引进 π_0 和 π_{n+1} 作为是假定的开始（begin）和结束（end）状态。

这个模型对给定的序列 x 和给定的路径 π 定义了概率 $P(x\mid\pi)$。然而只有这个商人知道发出 x 的状态 π 的真实序列。因此可以说 x 的路径是隐藏的并面临如下问题。

解码问题 对 x 寻找最优路径 $\pi^*=\underset{\pi}{\operatorname{argmax}}P(x\mid\pi)$，使得 $P(x\mid\pi)$ 最大。

解码问题的解决方法是由 Viterbi[348] 提出的，其算法是 Bellman[29] 的动态规划方法的一个变种。该方法的思想是，x 的 $i+1$ 个前缀 $x_1\cdots x_{i+1}$ 的最优路径就是在 x 的 i 个前缀中以（未知）状态 $\pi_i=k\in Q$ 结尾的路径中的最优的一个。

定义 $s_k(i)$ 为以状态 $k(k\in Q$ 和 $1\leqslant i\leqslant n)$ 结尾的前缀 $x_1\cdots x_i$ 最大可能的路径的概率，那么

$$s_l(i+1)=e_l(x_{i+1})\cdot\max_{k\in Q}\{s_k(i)\cdot a_{kl}\}$$

初始化 $s_{\text{begin}}(0)=1$ 和当 $k\neq\text{begin}$，$s_k(0)=0$。$P(x|\pi^*)$的值为

$$P(x|\pi^*)=\max_{k\in Q}s_k(n)a_{k,\text{end}}$$

Viterbi 算法所需运行时间为 $O(n|Q)$。在 Viterbi 算法中的计算经常使用对数得分 $S_l(i)=\lg s_l(i)$以避免计算溢出

$$S_l(i+1)=\lg e_l(x_{i+1})+\max_{k\in Q}\{S_k(i)+\lg(a_{kl})\}$$

给定投掷的序列 x，商人在 i 时刻使用有偏倚的硬币的概率是多少呢？Vitervi 算法的一个简单变异允许计算概率 $P(\pi_i=k|x)$。令 $f_k(i)$ 为发出前缀 $x_1\cdots x_i$和到达状态 $\pi_i=k$ 的概率。那么

$$f_k(i)=e_k(x_i)\cdot\sum_{l\in Q}f_l(i-1)\cdot a_{lk}$$

向前算法与 Viterbi 算法的仅有的区别在于 Viterbi 算法中的“最大”记号改变为向前算法中的 Σ 记号。向后概率 $b_k(i)$ 定义为在状态 $\pi_i=k$ 开始且发出后缀 $x_{i+1}\cdots x_n$ 的概率。向后算法使用类似的计算公式

$$b_k(i)=\sum_{l\in Q}e_l(x_{i+1})\cdot b_l(i+1)\cdot a_{kl}$$

最后，商人在 i 时刻使用有偏倚的硬币的概率为

$$P(\pi_i=k|x)=\frac{P(x,\pi_i=k)}{P(x)}=\frac{f_k(i)\cdot b_k(i)}{P(x)}$$

其中 $P(x,\pi_i=k)$ 是假设 x_i 在状态 k 和 $P(x)=\sum_{\pi}P(x|\pi)$ 产生的情况下 x 的概率。

8.9 Elkhorn 赌场和隐马氏模型参数估计

先前的分析假定隐马氏模型中的状态转换和发出概率是已知的。但在隐马氏模型的应用中最困难的问题是这些参数是未知的并且需要估计。一个聪明的赌徒很容易断定“公平赌场”中的商人正在使用一枚有偏倚的硬币。一种办法是揭发 0 和 1 出现的期望频率不同（分别为 3/8 和 5/8），且一整天投掷出的 0 和 1 之比可疑地低。但是，估计相应隐马氏模型中转换和发出的概率要困难得多。杰克·伦敦的作品《蹩脚·贝路》（Smoke Bellew）中的斯莫克最先尝试着估计 Elkhorn 赌场中轮盘赌的转换概率。经过长时间观察轮盘赌，一天晚上斯莫克宣称他可以胜过它。但是不要在拉斯维加斯再试着这样做，因为赌博技术早已改变。

设 Θ 是一个组合了隐马氏模型 $\mathcal{M}$ 的未知转换和发出概率的向量。给定一个字符串 x，定义 $P(x|\Theta)$ 为给定参数 Θ 时 x 的概率。目标是找出 Θ^* 使得

$$\Theta^*=\arg\max_{\Theta}P(x|\Theta)$$

通常，给定一个训练序列的集族 $x^1,\cdots,x^m$ 代替单个字符串 x，目的是使

$$\Theta^*=\arg\max_{\Theta}\prod_{j=1}^{m}P(x^j|\Theta)$$

最大。这是连续函数在多维参数空间 Θ 中的最优化问题。对于参数最优化，通常使用的算法是在参数空间中利用局部改进策略的试探法。如果相应于观测状态 $x_1,\cdots,x_n$ 的路径 $\pi_1,\cdots,\pi_n$ 是已知的，那么就可以仔细地查看序列，计算出转换和发出概率的经验估计。如果 A_{kl} 是从状态 k 到 l 的转换次数，$E_k(b)$ 是 b 从状态 k 发出的次数，那么最大似然估计为

$$a_{kl}=\frac{A_{kl}}{\sum_{q\in Q}A_{kq}},\quad e_k(b)=\frac{E_k(b)}{\sum_{\sigma\in\Sigma}E_k(\sigma)}$$

通常，状态序列 $\pi_1,\cdots,\pi_n$ 是未知的，在这种情况下，一般使用称为 Baum-Welch 算法[24]的迭代局部改进策略。

8.10 剖面隐马氏模型联配

给定一族功能相关的生物序列，通过将此族序列与数据库中序列成对联配，可以搜索此序列家族中的新成员。但是，这个方法可能对关系较远的相关序列无能为力。而迭代方法在作搜索时利用了整个功能相关序列集。

一个剖面是一族相关蛋白质最简单的表示法，这族蛋白质由多重联配给定。

给定 n 列字符串的多重联配，其中字符串的字符取自字母表 $\mathcal{A}$，剖面 $\mathcal{P}$ 是一个 $|\mathcal{A}|\times n$ 的矩阵，指定了每个取自字母表 $\mathcal{A}$ 的字符 a 在 i 列的频率 $e_i(a)$[139]。剖面之间可以相互比较和联配，这是因为如果两条输入序列均为多重联配，则用来联配两条序列的动态规划算法同样可行[362]。

隐马氏模型也可用于序列比较[208,324]，特别是一条序列对一个剖面进行联配。对于剖面 $\mathcal{P}$ 最简单的隐马氏模型包含了 n 个连续相连的匹配状态 M_1，…，M_n，其发出概率 $e_i(a)$ 取自剖面 $\mathcal{P}$（见图 8.3）。给定剖面 $\mathcal{P}$ 的字符串 x_1，…，x_n 的概率为 $\prod_{i=1}^{n} e_i(x_i)$ 。为了模拟插入和缺失，在隐马氏模型中增加了插入状态 $I_0,\cdots,I_n$ 和缺失状态 $D_1,\cdots,D_n$，并假设

$$e_{I_j}(a)=p(a)$$

其中 $p(a)$ 是符号 a 在所有序列中出现的频率。在仿射缺口罚分模型（the affine gap penalty model）中，通过指派 a_{MI}，a_{IM} 和 a_{II}，使得 $\lg(a_{MI})+\lg(a_{IM})$ 等于缺口产生罚分及 $\lg(a_{II})$ 等于缺口扩展罚分，从而可定义匹配和插入状态间的转换概率。（沉默）缺失状态不发出任何符号。

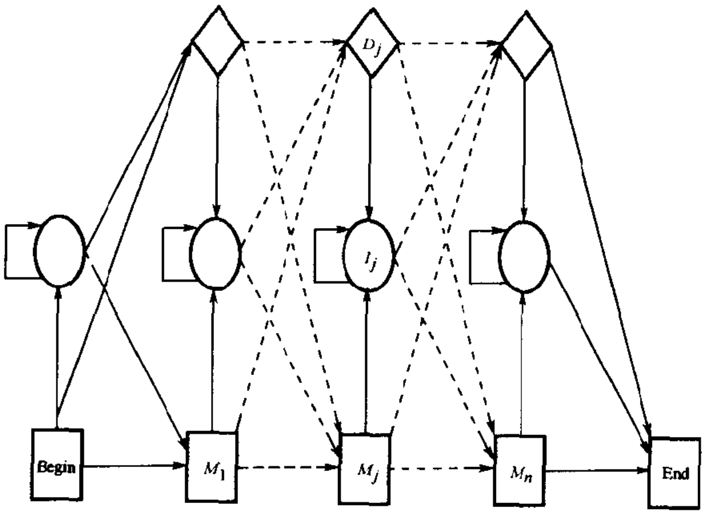


图 8.3 剖面隐马氏模型

对于以 x_i（由状态 M_j 发出）为终止的剖面隐马氏模型 $\mathcal{P}$，定义 $v_j^M(i)$ 为匹配 $x_1,\cdots,x_n$ 的最佳路径的对数似然得分。类似地定义 $v_j^I(i)$和 $v_j^D(i)$。结果，相应的动态规划循环非常类似于标准联配循环

$$v_j^M(i)=\lg\frac{e_{M_j}(x_i)}{p(x_i)}+\max\begin{cases}v_{j-1}^M(i-1)+\lg(a_{M_{j-1},M_j})\\ v_{j-1}^I(i-1)+\lg(a_{I_{j-1},M_j})\\ v_{j-1}^D(i-1)+\lg(a_{D_{j-1},M_j})\end{cases}$$

$v_j^I(i)$和 $v_j^D(i)$的值可类似地定义。

8.11 吉布斯抽样

Lawrence 等[217]提出利用吉布斯抽样来找出序列中的模式。给定一组字母表 A 中的序列 $x^1,\cdots,x^m$ 及整数 w，问题是在每一个 x^i 中找出长度为 w 的子序列，使得这 m 个子序列间的相似度为最大。

设 $a^1,\cdots,a^m$ 分别是这些从 $x^1,\cdots,x^m$ 中选出的子序列的起始点。用 q_{ij} 表示符号 i 出现在子序列中第 j 个位置的频率。

吉布斯采样是个迭代过程，每一步迭代从联配中去除一条序列，并以一条新的序列来替代。在 m 个序列 $x^1,\cdots,x^m$ 的每一个中任取一长度为 w 的子序列，开始吉布斯采样，其过程如下。

- 每步迭代开始时，在 m 个序列 $x^1,\cdots,x^m$ 的每一个中任取一长度为 w 的子序列。
- 均匀随机地选取一条序列 x^r。
- 用余下的 $m-1$ 条子序列产生一个频率矩阵 (q_{ij})。
- 对 x^r 中的每个位置 i，计算由剖面 (q_{ij}) 产生的子序列从此位置开始的概率 $p_i=\prod_{j=0}^{w} q_{x^r_{i+j},j}$（$x^r_{i+j}$表示位于序列 x^r 中 $i+j$ 位置上的符号）。
- 在 x^r 中随机地选择新序列的起始位置 i，其概率和 p_i 成比例。

尽管吉布斯采样对特殊的情况很有效，但类似于 Baum-Welch 算法，它可能收敛于局部最大。由于所描述的过程[217]不允许插入和缺失，Rocke 和 Tompa[288]推广了此方法用于处理模式中的缺口。

8.12 若干其他的问题和方法

8.12.1 找出带缺口的信号

Rigoutsos 和 Floratos[285]研究了在文本中找出带缺口的信号的问题。带缺口的字符串是这样的一条字符串，它由字母表中的符号和“无关”符号任意组合而成的。一个带缺口的字符串 P 称为 $\langle l,\ w\rangle$-字符串，如果 P 中每个长度为 l 的子字符串至少包含字母表中 w 个符号。TERESIAS 算法[285]可找出所有最大的 $\langle l,\ w\rangle$-模式，它们至少出现在两阶段方法中的样本的 K 条序列中。在第一个扫描阶段，它找出所有长度为 l，且带有至少 w 个符号的短字符串，这些符号来自字母表，且在样本中至少出现 K 次。在第二个卷积阶段，它将这些短字符串装配成最大 $\langle l,\ w\rangle$-字符串。

8.12.2 在带偏倚频率样本中找出信号

如果信号仅包含在所有序列的小部分中，且样本中的背景核苷酸分布有偏

差，则魔力词问题将变得难解。在这种情况下，搜寻带最大出现数的信号可能通向一个由最频繁核苷酸组成的模式。而这些模式在生物学上可能不怎么重要。例如，若 A 出现频率为 70%，T，G，C 的出现频率为 10%，则 poly（A）可能为最常出现的词，这样可掩盖真正的魔力词。

为了在带偏差的样本中找出魔力词，许多算法用相对熵使魔力词在从最常出现的核苷酸组成的词中突出出来。给定一长度为 k 的魔力词，相对熵定义为

$$\sum_{j=1}^{k} \sum_{r=\mathrm{A,T,G,C}} p_{rj} \log_2 \frac{p_{rj}}{b_r}$$

其中 p_{rj} 是魔力词出现时核苷酸 r 在位置 j 的频率，b_r 是 r 的背景频率。

对于比较在样本中有相同出现次数的两个魔力词，相对熵是一个好的度量，但对出现的序列数目相差巨大的魔力词，相对熵不是个好的度量。Hertz 和 Stormo[159] 及 Tompa[338] 论述了这个问题，他们设计了既处理出现次数也处理背景分布的标准。

8.12.3 信号寻找中字母表的选择

Karlin 和 Ghandour[187] 注意到没有办法预先知道选择哪个字母表对揭示 DNA 或蛋白质中的信号有利。例如，WSWS…WSWS 是一个很强的信号，它在标准 A、T、G、C 字母表中很难找到。Sagot 等[293]论述了这个问题。

9 基因预测

9.1 引言

20世纪60年代，Charles Yanofsky、Sydeny Brenner和他们的合作者指出基因和它的蛋白质产物有着线性对应结构，基因中核苷酸三联体与蛋白质中氨基酸之间有直接的相关性。然而，将基因作为核苷酸合成序列的观点没有存在很长时间。20世纪60年代末，发现了重叠基因和内含基因。这些研究证明基因预测的计算问题远非简单。最后，1977年人类断裂基因的发现引起了一个计算基因预测难题。

真核生物基因组和原核生物基因组相比更大、更复杂。既然人们期待在人体内发现比细菌中更多的基因，这就没什么惊奇了。但是，一些真核生物基因组的大小并没有呈现出与遗传复杂性之间的联系：例如，火蜥蜴的基因组比人的基因组大10倍。由于发现了真核生物不仅包含基因还包含大量不编码任何蛋白质的DNA（“垃圾”DNA，“junk”DNA），这个似非而是的论点才得到解答。而且，多数人类基因被垃圾DNA割断，并且插入外显子（exon）的片段。火蜥蜴与人类基因组大小的差异反映出火蜥蜴基因组中存在大量的垃圾DNA和重复片段。

1977年，Phillip Sharp实验室与Richard Roberts实验室在对腺病毒的研究过程中首次独立地发现了断裂基因[32,67]。这个发现是如此神奇以致Richard Roberts小组在投给细胞杂志的论文中使用一个异乎寻常且令人费解的标题：“An amazing sequence arrangement at the 5’ end of adenvirus 2 messenger RNA.” Berget等[32]把实验重点放在一条可编码成已知六邻体病毒蛋白质的mRNA上。为了在病毒基因组上制作此六邻体mRNA的图谱，将mRNA与腺病毒DNA杂交，使用电子显微镜分析杂交分子。值得注意的是，在这个实验中形成的mRNA-DNA杂交产物显示出3个环状结构，不同于经典的“连续基因”模型认为的连续双螺旋区段。更多的细胞实验揭示这个六邻体mRNA是由腺病毒基因组上4个分离片段构成的。腺病毒基因组中的这4个外显子被3个称为内含子（intron）的垃圾片段所分开。腺病毒中断裂基因（剪接）的发现很快就出现了其他证据，哺乳动物的基因也有断裂结构[337]。这些实验研究再次提出了仍旧未被解决的计算基因预测问题：人类基因仅包含人类基因组的3%，而现有的计算机识别算法并不提供可靠的基因识别。

一条新的DNA测序完成后，生物学家努力从中寻找基因。处理这个问题的传统的统计学方法是寻找在基因中频繁出现的而在别处却罕见的特征。许多研究

者运用较多生物学导向的方法试图识别在外显子-内含子接头附近剪接信号的位置。这种方法的目标是识别对 RNA 上涉及剪接机构的蛋白质和核糖核蛋白的结合/相互关系的位置的表示特征。例如，在外显子左右两侧的二核苷酸 AG 和 GT 是高度保守的。描述一个信号的最简单方法是给出一条由在特殊信号联配中每个位置上频繁出现次数最多的核苷酸组成的共有序列模式。虽然在 20 世纪 80 年代初，剪接位点目录已被编辑完成。但对于从假位点中区分正确位点，共有序列模式并不是非常可靠的，因为它们不包含在不同位置的核苷酸频率的信息。Ed Trifonov 给出了一个说明共有序列另一个潜在缺陷的例子。

MELON
MANGO
HONEY
SWEET
COOKY
——
MONEY

剖面（或位置权重矩阵）记录频率信息，对每一个符号位置上的每一个可能核苷酸分配一个基于频率的得分。不幸的是，可能由于多结合分子之间协同作用，利用剖面方法做剪接位置预测只取得了有限的成功。出于提高基因预测精确性的目的，神经网络模型和隐马氏模型开始应用于基因发现。

大规模测序计划激发了新一代基因识别算法的需求。一条新测序的基因有机会在数据库中得到一条已知的亲缘序列，在此观察基础上建立了相似性的基因预测方法[41]。如潮水般增长的新的测序数据会很快使这种可能性非常大，其结果，在 20 世纪 90 年代后期，基因预测趋势从基于统计学的方法转向基于相关性的算法和基于 EST 的算法。特别，Gelfand 等[125]提出了一种基因预测的组合方法，利用相关蛋白质得到外显子-内含子结构。取代使用关于外显子的统计学特性，这种方法试图解决一个组合难题：在基因组序列中寻找一个子序列集合，这些子序列串联（剪接）起来适合已知蛋白质中的某一条。

在做出预测以后，生物学家尝试通过实验检验它们，检定通常等同于全长 mRNA 测序。由于这个过程是相当耗费时间的，计算机预测在设法将结果装入数据库时会频繁地出现注释错误。仅仅能够猜测 GenBank 中不正确注释的序列的数量，但很清楚的是，除全长 mRNA 数据（和有潜在错误的）外，已注释过的基因数量是很大。发展一个“快速注释”（annotation-ready）的基因预测算法并同时校正这些错误的问题依旧未被解决。

9.2 基因预测的统计学方法

发现潜在编码区域的最简单的方法是查看开读框（Open Reading Frame,

ORF)。一个 ORF 是 DNA 中以起始密码子为起点，以终止密码子为终点的密码子序列，其中没有其他的终止密码子。人们指望在非编码 DNA 中简单地找到频繁的终止密码子，因为 64 个可能密码子中的 3 个是翻译终止子。在“随机”DNA 序列中终止密码子之间的平均距离是 64/3≈21，与蛋白质的平均密码子数目（大概 300）相比要小得多。因此，长的 ORF 指出了潜在的基因[105]，虽然它们未能探测出短的基因或具有短外显子的基因。

许多基因预测预算法依靠识别蛋白质编码区域中的扩散规则，例如对密码子的选择性。密码子选择是在一个窗口内 64 个可能密码子（核苷酸三联体）的每一个的发生频率的 64 位向量。密码子选择向量在编码窗口和非编码窗口间是不同的，这就使人们能够利用这种方法来进行基因预测[104,327]。Gribskov 等[138]在编码和非编码的随机序列假设下，使用似然比方法计算一个窗口内 DNA 序列的条件概率。当窗口沿 DNA 滑动时，基因经常展现为似然比曲线的波峰。好的编码检测器是符合读框的六聚体计数（in-frame hexamer count），类似于 3 个五阶马尔可夫模型[42]。Fickett 和 Tung[106]评价了许多这样的编码方法并且得到一个结论：由于存在许多假阳性和假阴性的排布，它们给出了一个编码区边界的相当低分辨率的图片。此外，这些技术对真核生物的运用由于外显子-内含子结构的存在而变得复杂。在脊椎动物中外显子的平均长度是 130bp，这样外显子经常太短以至不能在滑动窗口曲线中产生波峰。

密码子的选择、氨基酸的使用、编码区的周期和其他统计参数[123]可能与剪接机器识别外显子的方法没有共性可言。许多研究人员使用较多生物学导向的方法并设法识别在外显子-内含子接合处剪接信号的位置[50]。在外显子和内含子边界上（5′端或剪接供体）存在一条 8 个核苷酸的（弱）保守序列，且在内含子和外显子边界（3′端或剪接受体）上有一条 4 个核苷酸的序列。不幸的是，可能是由于多结合分子间的相互作用的缘故，用于剪接位置预测的剖面方法只取得了有限的成功。剖面方法相当于一种称为感知器（perceptron）的简单的神经网络。更复杂的神经网络[339]和隐马氏模型[209]已被应用。Burge 和 Karlin[54]通过捕捉这些位点之间的统计相关性改进了预测质量。

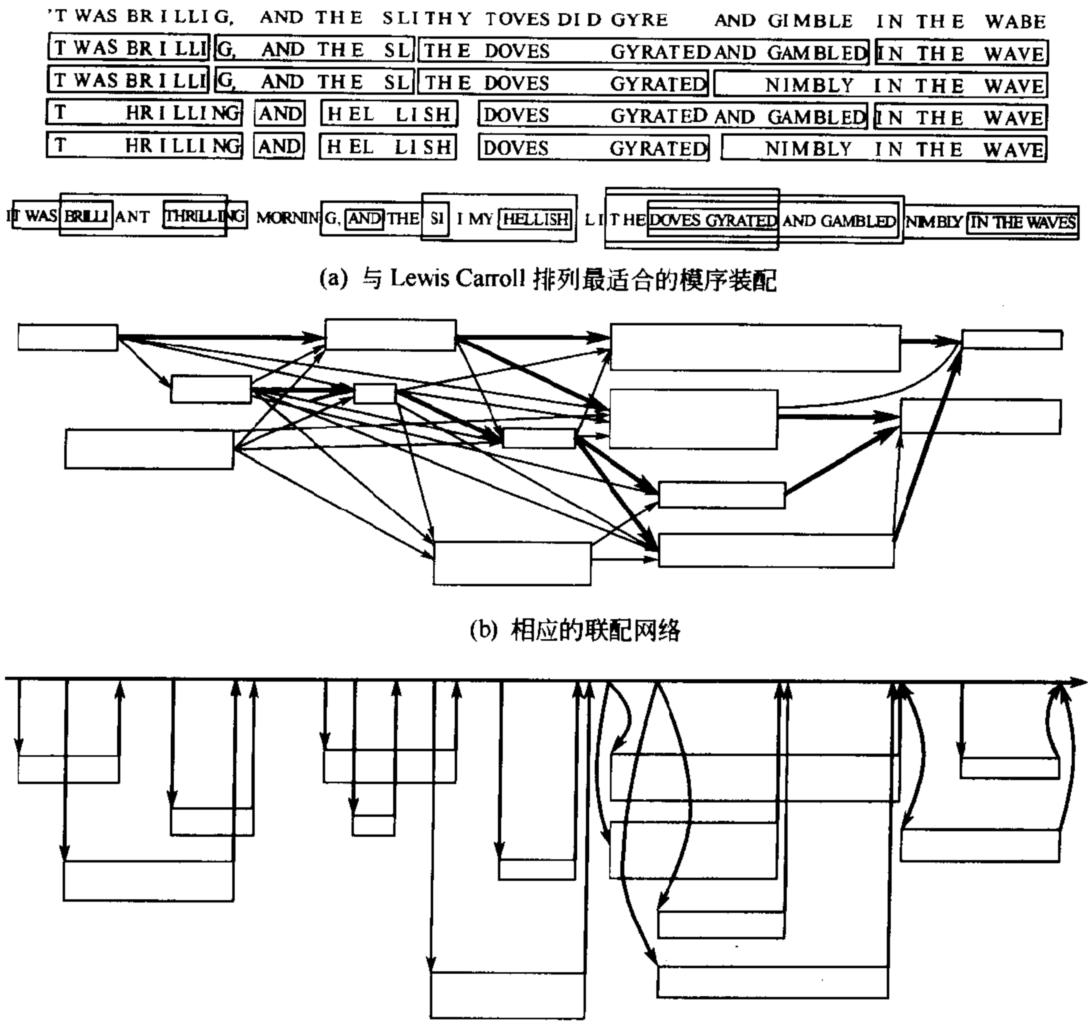
许多研究人员试图把编码区和剪接信号预测并入一个信号框架之中。例如，如果一个编码区的信号出现在一个剪接位点的一侧而不是另一侧，那么这个剪接位点预测有较高可信度。由于个体统计量的限制，一些研究小组已发展出将多个评估判据结合进单个框架的基因预测算法[121,142,249,321]。事实上，对于 GENSCAN[54]的隐马氏模型框架而言，使用了全部已有的统计量。这个算法不但结合了剪接位点、启动子、聚腺苷酸化位点和编码区的统计量，而且考虑它们的非均匀性。这就让作者们超越了基因预测的统计方法可以达到的 90%精确度的里程碑。然而，值得注意的是，对于具有许多短的外显子或稀有密码子的基因，方法

的精确度反而显著降低。

9.3 基于相似性的基因预测方法

基于相似性的基因检测方法的思想由 Gish 和 States[129]第一次做了陈述。虽然相似性搜索用于基因检测（也就是回答一个基因是否在一条给定的 DNA 片段中出现的问题）已有较长时间了，但是用于基因预测的相似性搜索的潜力（即不仅用于基因检测，而且也对外显子-内含子结构进行详细的预测）直到 20 世纪 90 年代中期仍有许多未曾仔细研究。Snyder 和 Stormo[322]以及 Seals 和 Murphy[313]第一次尝试将相似性分析合并入基因预测算法。然而，由于要对序列联配算法顶部所有的外显子装配进行检测，计算复杂性是很高的。

Gelfand 等[125]建议使用剪接联配方法解决外显子装配问题，即利用相关蛋白质推断外显子-内含子结构。图 9.1（a）阐明基因组序列的剪接联配问题。



(a) 与 Lewis Carroll 排列最适合的模序装配

(b) 相应的联配网络

(c) 联配网络的等价变换

图 9.1 剪接联配问题

It was brilliant thrilling morning and the slimy hellish lithe doves
gyrated and gambled nimbly in the waves

那些不同的模序（block）构成著名的 Lewis Carroll 排列

't was brillig, and the slithy toves did gyre and gimble in the wabe

Gelfand 等[125]的方法是基于如下的思想（图解由 Oksana Khleborodova 提供）。给定一条基因组序列（见图 9.2），他们首先找出一组包含所有正确外显子候选模序的集合（见图 9.3）。这可以通过选择在可能的接受位点和供体位点之间（即，在 AG 和 GT 二核苷酸之间）的所有的模序并对该集合作进一步过滤（在某种程度上不丢失真实的外显子）来实现。这些模序的结果集合可能包含许多错误的外显子，当然，目前利用统计方法还不可能从这个集合中辨别出所有真实的外显子。取代寻找真实外显子的尝试，Gelfand 等[125]从 GenBank 中挑选一个相关的靶蛋白（见图 9.4），然后以发现一个与靶蛋白具有最高相似性得分的



图 9.2 研究基因组序列



图 9.3 过滤候选外显子



图 9.4 寻找靶蛋白



图 9.5 利用靶蛋白作为外显子装配的模版

装配为目标，探测所有可能的模序装配（见图 9.5）。不同模序装配的数目是巨大的（见图 9.6～图 9.8），但这个算法中的关键部分剪接联配算法可在多项式时间内扫描所有集合（见图 9.9）。



图 9.6 装配



图 9.7 继续装配



图 9.8 再继续装配

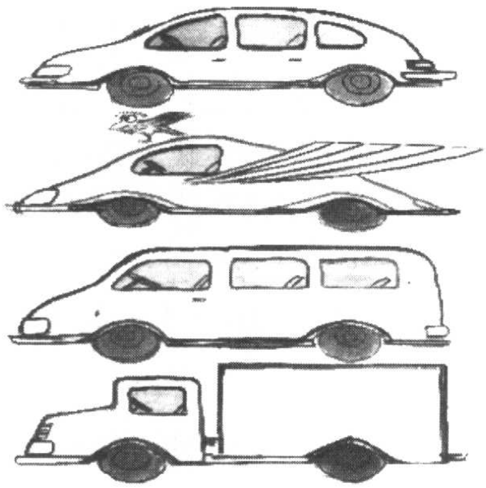


图 9.9 选择最佳外显子装配

9.4 剪接联配

设字符串 $G=g_1\cdots g_n$，同时记 $B=g_i\cdots g_j$，$B'=g_{i'}\cdots g_{j'}$ 是 G 的子字符串。如果 $j<i'$，也就是 B 在 B' 起始前终止，则记 $B\prec B'$。如果 $B_1\prec B_2\prec\cdots\prec B_p$，则称 G 的子字符串的序列 $\Gamma=(B_1,\cdots,B_p)$ 为链。记链 Γ 中的字符串的串联为 $\Gamma^*=B_1*B_2*\cdots*B_p$。给定两条字符串 G 和 T，记 $s(G, T)$ 为 G 和 T 之间最佳联配的得分。

设 $G=g_1\cdots g_n$ 是称为基因组序列的字符串，$T=t_1\cdots t_m$ 是称为靶序列的字符串。G 的子字符串集合记为 $\mathcal{B}=\{B_1\cdots B_b\}$，称为模序（block）。给定 G，T 和

$\mathcal{B}$，剪接联配问题即从 $\mathcal{B}$ 中寻找一条字符串的链 Γ，使得这些字符串的串联和靶序列之间的联配得分 $s(G^*, T)$ 在所有属于 $\mathcal{B}$ 的模序的链中达到最大。

解决剪接联配问题的自然方法是检测在每个模序和靶序列之间所有相对高相似处并且把这些相似处装配成一致相似片段的最佳子集。这种方法的缺点是模序的数目在通常情况下是非常大的而相似域的终点也很难定义。

Gelfand 等[125]将外显子装配问题归结为在方向图上搜索路径的问题［见图 9.1 (b)］。图 9.1 (b) 中的顶点对应于模序，边对应模序之间可能的转移，同时路径的权被定义为在这条路径所连接的所有模序与靶序列之间最佳联配的权。注意外显子装配问题与标准最短路径问题不同［顶点的权和图 9.1 (b) 中的边无法恰当定义］。

记 $B_k = g_m \cdots g_i \cdots g_l$ 是 G 中包含位置 i 的子字符串。定义 B_k 的 i 前缀为 $B_k(i) = g_m \cdots g_i$。对于模序 $B_k = g_m \cdots g_l$，记 $first(k) = m, last(k) = l, size(k) = l - m + 1$。记 $\mathcal{B}(i) = \{k : last(k) < i\}$ 为 G 中在位置 i 之前终止（严格）的模序集合。包含位置 i 的一些模序 B_k 所构成的链记为 $\Gamma = (B_1, \cdots, B_k, \cdots, B_t)$。定义 $\Gamma^*(i)$ 为字符串 $\Gamma^* = B_1 * B_2 * \cdots * B_k(i)$。记

$$S(i,j,k) = \max_{\Gamma \in B_k} s(\Gamma^*(i), T(j))$$

接下来在 $1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m, 1 \leqslant k \leqslant b$ 范围内重复计算 $S(i,j,k)$。出于简便的缘故，考虑具有线性缺口罚分的序列联配，且定义 $\delta(x, y)$ 为每对氨基酸 x 与 y 的相似性得分，δ_{indel} 则为氨基酸插入或缺失的罚分。

$$S(i,j,k) = \max \begin{cases} S(i-1, j-1, k) + \delta(g_i, t_j), & i \neq first(k) \\ S(i-1, j, k) + \delta_{\text{indel}}, & i \neq first(k) \\ \max\limits_{l \in \mathcal{B}(first(k))} S(last(l), j-1, l) + \delta(g_i, t_j), & i = first(k) \\ \max\limits_{l \in \mathcal{B}(first(k))} S(last(l), j, l) + \delta_{\text{indel}}, & i = first(k) \\ S(i, j-1, k) + \delta_{\text{indel}} \end{cases} \tag{9.1}$$

在计算完三维表格 $S(i,j,k)$ 后，最佳剪接链配得分是

$$\max_k S(last(k), m, k)$$

剪接联配问题也可以表述为网络联配问题[211]。在这种表述下，每一个模序 B_k 对应于在顶点 $first(k)$ 和 $last(k)$ 之间长度为 $size(k)$ 的一条路径，如果 $B_k < B_t$，则对应于模序 B_k 和 B_t 的路径通过边 $(last(k), first(t))$ 连接［见图 9.1 (b)］。网络联配问题就是在网络中寻找与靶序列最佳联配的路径。

Gelfand 等在 1996 年[125]使用对所述网络的等价变换减少剪接联配图中的边的数目，从而导致时间和空间上的简化。定义

$$P(i,j) = \max_{l \in \mathcal{B}(i)} S(last(l), j, l)$$

那么式（9.1）可改写为

$$S(i,j,k)=\max\begin{cases}S(i-1,j-1,k)+\delta(g_i,t_j), & i\neq first(k)\\ S(i-1,j,k)+\delta_{\text{indel}}, & i\neq first(k)\\ P(first(k),j-1)+\delta(g_i,t_j), & i=first(k)\\ P(first(k),j)+\delta_{\text{indel}}, & i=first(k)\\ S(i,j-1,k)+\delta_{\text{indel}} & \end{cases}\tag{9.2}$$

其中

$$P(i,j)=\max\begin{cases}P(i-1,j)\\ \max\limits_{k:last(k)=i-1} S(i-1,j,k)\end{cases}\tag{9.3}$$

值得注意的是，对应于式（9.2）和式（9.3）的网络具有较少的边数［见图9.1(c)］，这导致实际执行一种剪接联配的算法。

构造模序 $\mathcal{B}$ 的最简单方法是生成由 AG（接受位点）和 GT（供体位点）表示的可能剪接位点之间的所有片段，但不包括在所有三个读框中具有终止密码子的模序。然而，这种方法由于生成许多短的模序而产生一个问题。试验显示，剪接联配算法对关系远的靶的不正确预测频繁地与由非常短的可能外显子引起的镶嵌效应（mosaic effect）相联系。问题在于这些短的外显子可以容易地结合在一起去匹配任何靶蛋白。从许多随机的短字符串中“拼凑”一条给定序列要比在同样数目的长的字符串中生成相同的给定序列容易。例如，具有很高的可能性，短语“filtration of candidate exons”可由许多随机两字母字符串的样本拼凑而成（“fi”，“lt”，“ra”等，都可能在样本中出现）。但是，同一短语想要用相同数目的随机5个字母字符串的样本来拼凑，其概率几乎为零（即使在此样本中找到字符串“filtr”也不太可能）。这个观察解释了镶嵌效应：如果短的模序的数目很多，这些模序的链可以替换在剪接联配中真实的外显子，从而导致一个具有大量短外显子数目的预测。为避免镶嵌效应，利用某些（弱）过滤程序处理候选外显子；例如，只保留具有高编码可能性的外显子。

最佳模序装配发现以后，希望它表示正确的外显子-内含子结构。如果一个蛋白质充分相似于所分析片段中编码的一个蛋白质，那么几乎可保证是有用的：与真实基因具有99%相关性的基因可以从具有100 PAM距离（40%相似）的靶中获得。剪接联配算法提供了非常精确的预测，即使一个远缘蛋白质也可利用：在160 PAM（25%相似）的预测仍是可信的（95%的相关性）。对于一条已经分析过的人类基因，如果已知一条与之相关的哺乳动物蛋白质，那么在这个片段中基因预测的精确度可高达97%～99%，而对相关的植物，真菌，原核生物蛋白质的预测精确度则分别可达到95%，93%和91%[242]。基于相似性的基因预测方法由于利用了EST数据，使其在基因预测方面取得了更多的进展。特别是，通过使用EST装配，Mironov等[240]发现了大量可变剪接基因。

9.5 反向基因搜索和 cDNA 中外显子定位

基因搜索经常遵循定位克隆范例：基因组 DNA 测序⟶外显子结构⟶mRNA⟶蛋白质。

在定位克隆工程中，基因组的序列是基因预测，突变检测和致病基因进一步搜索的主要资源。从定位克隆到候选基因库的策略是反向的，传统的基因搜索途径如下所示。

蛋白质/mRNA⟶外显子结构⟶有限的基因组 DNA 测序。

因此，现代基因发现的研究工作正从单个基因的定位克隆向具有数百个基因构成候选基因文库的多基因疾病的解析转变。构成候选基因文库的基因可以来自不同的资源，例如，表达分析、抗体筛选、蛋白质组学等。当然，克隆数百个基因太昂贵以至于无法实际应用。

定位克隆方法是通过遗传作图来发现引发疾病的致病基因，并进行疾病相关突变的检测。这需要实施多个步骤，包括大 DNA 片段的基因组克隆、cDNA 文库筛选、cDNA 分离、为测序进行的大的基因组克隆的亚克隆等（见图 9.10）。在许多基因搜索的努力中，基因组亚克隆和测序步骤的主要动机是决定基因外显子-内含子的边界。此步骤对致病基因相关连的突变（或多态性）的搜索常常是关键性的。它要求在每个外显子的两侧设计内含子的 PCR 引物。传统上，外显子边界可通过 cDNA 序列与基因组序列的比较而得到。整个过程不仅耗费时间，而且包括多个亚克隆步骤及广泛的测序。

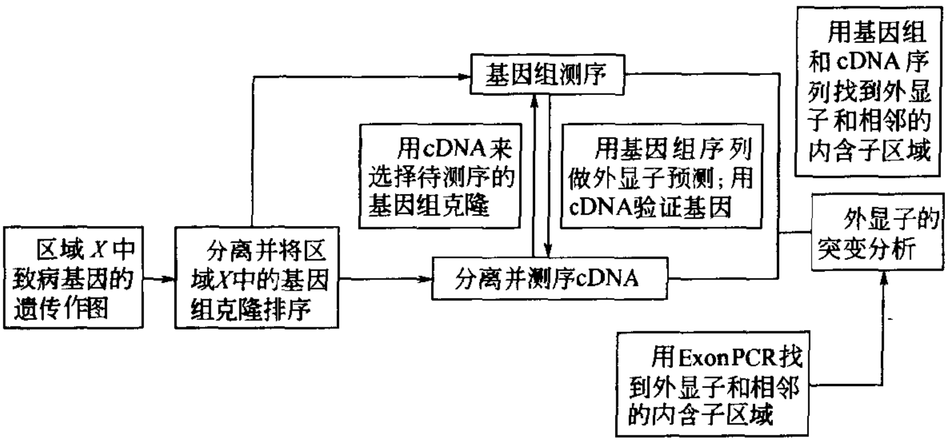


图 9.10 定位克隆和 ExonPCR

ExonPCR[372]是一个探索“反向”基因发现途径并提供从 cDNA 搜索到突变检测的快速转变的交错实验操作规程（见图 9.10）。ExonPCR 搜寻 cDNA 中（还不如说在基因组 DNA 中）“隐藏”的外显子边界，但不要求基因组克隆测序。第一步，ExonPCR 在基因组 DNA 上利用根据 cDNA 序列设计的引物通过

PCR 方法定位外显子边界在 cDNA 中的近似位置。第二步，通过执行连接-介导 PCR 寻找内含子区域的两侧，其结果，DNA 测序工作量可极大地减少。

发现 cDNA 中外显子边界的计算方法[122,323]搜索剪接阴影（splicing shadows）（即在外显子中剪接信号出现的部分）。然而，由于剪接阴影信号非常弱，因此相应的预测是不可信的。ExonPCR 是一种实验方法，它利用 PCR 引物，通过适当次数的重复实验，在 cDNA 中寻找外显子的边界。引物根据 cDNA 序列来设计，并用于扩增基因组 DNA。每一对引物用于回答一个质疑问题：在基因组 DNA 内，在引物序列之间是否存在一个或多个内含子。通过比较 cDNA 中 PCR 产物长度和基因组 DNA 长度可以回答这个疑问。如果两者长度相等，则引物属于同一外显子；否则，在对应引物之间存在一个外显子边界。每一对引物仅给出一个是/否的答案，而没有给出外显子边界的精确位置。目的是为了寻找外显子边界的位置，同时使引物数目和实验次数都最少。为此可以使用不同类型的策略，且此问题类似于基因的“Twenty Qusetions”（20 个问题）对策。它和室内游戏（parlor game）对策不同之处在于基因有一个“no answer”选项，有时或许给出一个错误的答案并且限制可能疑问的类型。这类似于“Twenty Questions Game with a Liar”（说谎者与 20 个问题的游戏）[216]，但涉及许多附加的约束，包括有关查询长度（PCR 引物之间的距离）的下界和上界。

ExonPCR 试图设计一种策略：使得需要的 PCR 引物的总数最少（减少花费），同时使 PCR 实验必须的重复次数最少（减少时间）。然而，这两个目标是彼此冲突的。最小数目引物对可按连续的“二分法（dichotomy）”操作规程得到，在每次实验中都在前一轮实验结果的基础上仅设计一对引物。由于这个策略导致过多的实验次数，因此它是不现实的。作为一个替代，“一次性”的操作规程则在一次实验中设计所有可能的引物对，但这导致了过多的引物数量。由于这两个判据冲突，ExonPCR 在二分法策略和一次性策略之间寻找一个平衡。

9.6 关于基因的多次提问策略

依它的最简单形式，此问题能够表述为：在区间 $[1, n]$ 内，给定一个（未知）整数集合 I，通过最少次数形如“一个给定区间包含集合 I 中的整数吗?”的查询来重构集合 I。在这种表述下，区间 $[1, n]$ 对应于 cDNA，I 对应于 cDNA 中外显子边界，而查询则对应于由一对引物确定的 PCR 反应。这个问题的非自适应（平凡的）方法产生 n 个单点（single-position）查询：区间 $[i, i]$ 包含 I 中的一个整数吗? 在自适应方法中，根据以前全部查询的结果（每次循环仅产生一个查询）产生当前循环的查询。

为简便的缘故，考虑外显子边界数目 k 已知的情形。当 $k=1$ 时，这个问题的最佳算法要求至少 $\lg n$ 次查询，且类似于二进制搜索[75]。当 $k>1$ 时，容易得

到对此问题使用任何一种与决策树模型有关的算法所需要的查询次数的下界。决策树模型假定连续的计算每次使用一个查询。假定决策树上每个顶点与所有和在这顶点路径上全部查询相容的 k 点集合（k-sets）相联系。由于决策树上的每片叶子只包含一个 k 点集合，因此叶子的数目是$\binom{n}{k}$。由于每棵高 h 的树至多有 2^h 片叶子，因此（二进制）决策树的高度的下界是 $h \geqslant \lg \binom{n}{k}$。在生物学上相关情形 $k \ll n$，查询的最小数目近似为 $k\lg n - k\lg k$。如果生物学家容许在外显子边界定位上有误差 Δ，那么查询数目的下界近似为 $k\lg \frac{n}{\Delta} - k\lg k$。ExonPCR 计算和实验测试证明它接近理论下界，同时在一条典型的 cDNA 序列中判明外显子边界需要大约 30 个引物和 3 次循环。

9.7 可变剪接与癌症

最近的研究为可变剪接能够调节人癌中的致癌潜力提供了证据。例如，前列腺癌从对男性激素敏感的肿瘤到对男性激素不敏感的肿瘤的发展伴随着成纤维细胞生长因子受体 2 上可变剪接的改变[59]。在另一项研究中，Heuze 等[160]描述了一个对前列腺专一抗原显著的可变剪接变体，它成为目前诊断和监测前列腺癌病人最重要的可利用标记。有关这些基因及其他基因的其他重要的剪接变体的问题在肿瘤研究中仍然是未解决的。而且，这个已知的肿瘤中的基因的可变变体是通过病例对病例（case-by-case）方式的检查而被偶然发现的。

给定一个基因，如何找到这个基因的所有可变剪接变体？由于可变剪接在人类基因中非常频繁地出现[240]，因此这个问题远非简单，而且可变剪接预测的计算方法也是非常不可靠的。

Hu 等[167]第一次系统地尝试说明包含在（卵巢）癌中的基因的剪接变体。他们提议使用长的 RT-PCR 扩增全长 mRNA，并且找到了一个新的人类多抗病基因 MDR1 和主要穹状蛋白质（MVP）的剪接变体。这个方法非常适合检测少量使用固定引物的显著变体，但在检测稀有变体方面则有困难（由于显著变体不受抑制）。如不用选定的引物对扩增，它也不能识别显著剪接变体。

寻找全部可变剪接变体的计算的复杂问题（一本可变剪接百科全书即 ASE）可以用下述例子予以解释。如果一条含有 3 个外显子的基因具有一个失去中间那个外显子的可变变体，那么根据这个中间外显子的长度可以区分 cDNA 文库中一些 PCR 产物。例如，一对引物，其中之一来自第一个外显子的中部、另一个来自最末一个外显子的中部，它们会给出被中间外显子长度区分的两个 PCR 产物。这会引起对双方可变剪接变体的检测。

当然，这是一个仅仅出于说明目的的简单而又自然的叙述。如果考虑一个含

有 10 个外显子且每个外显子中都有一个可变的不稳定剪接位点，那么这个问题的复杂性就可理解了。在这种情况下，潜在剪接变体的数目至少是 2^{10}，并且不清楚如何去找到细胞中存在的变异体。实际的问题甚至更加复杂，因为这些剪接变体中的一部分可能是稀有的并且很难通过 PCR 扩增进行检测。

图 9.11 说明为“基因组”序列

'twas brilliant thrilling morning and the slimy hellish lithe doves

Gyrated and gambled nimbly in the waves

构建一个 ASE 问题，而可变剪接变体“编排”了不同的 mRNA，它类似于 Lewis Carroll 的著名的“mRNA”

't was brillig, and the slithy toves did gyre and gimble in the wabe

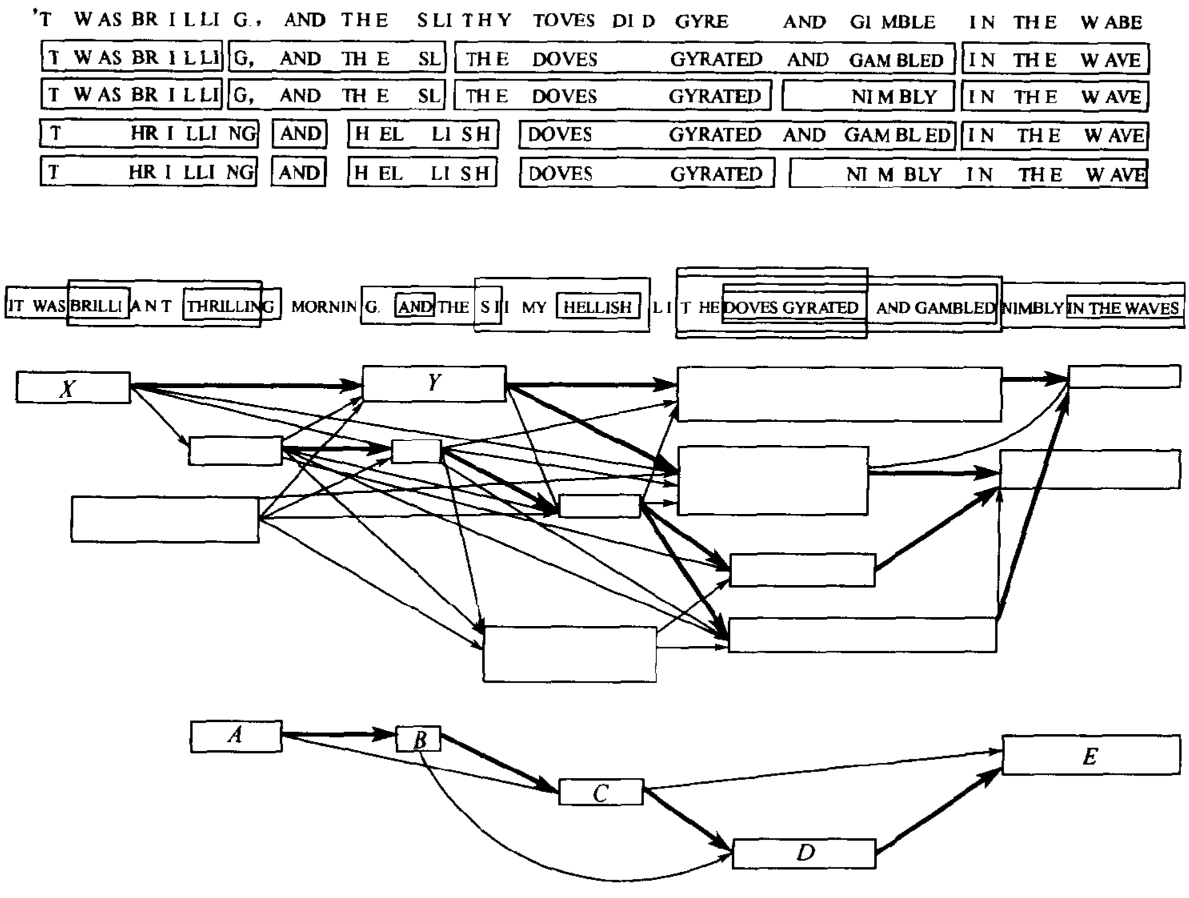


图 9.11 从潜在的外显子中构造一个可变剪接百科全书（ASE）

在这个外显子装配图中 4 个不同的剪接变体（上面）对应 4 条路径（粗边显示）。这张图上全部的路径数目是巨大的，问题是如何识别对应于真正剪接变体的路径。这张图实际上描绘出一张整理过的刚好有 5 个潜在剪接变体（路径）的外显子装配图

“外显子装配”图（见图 9.11）中的路径数呈指数形式，每一条路径表示一个可能的剪接变体。问题是断定哪一条路径对应真正的剪接变体。例如，可以检

查是否存在剪接变体，它通过跨越 X 和 Y 的跨越引物 XY（例如，BRILLIG，AND T）把由 T WAS BRILLI 和 G，AND THE SL 表示的潜在外显子 X 和 Y 结合起来。在实践中，通过串联外显子 X 尾部的 10 个核苷酸和外显子 Y 头部的 10 个核苷酸来构建引物 XY。与其他引物的配对 XY（例如，取自外显子 Y 的尾部）将确认或否决关于结合外显子 X 和 Y 的剪接变体存在性的假设。跨越引物允许修整没有被实验证据支持的外显子装配图的边界。即使图的一些边界修整后，此方法仍需面对一个困难问题，即决定可能出现在可变剪接基因之间的外显子的三倍体、四倍体等。图 9.11 介绍了一个相对简单的例子，这是一张修整好的外显子装配图，有 5 个潜在外显子和 5 条可能路径：$ABCDE$，$ACDE$，$ABDE$，$ABCE$ 和 ACE。变异体 ACE 的惟一跨越引物对是 AC 和 CE。然而，这些跨越引物对（在与一些其他引物配对中）不允许我们去确认或排除 ACE 剪接变体的存在。其理由是，比方说被一个包含 AC 的引物对扩增的 PCR 产物的存在并不保证 ACE 变体的存在，因为这个产物可能来自 $ABCD$ 可变变体。同样，CE 引物可能扩增一个 $ABCD$ 剪接变体。如果幸运的话，可以观测到一个相对短的 ACE PCR 产物，但是，如果 ACE 是相对稀有的变体，这就不会发生。解决方案是构建一对包含 AC 和 CE 的跨越引物。这个引物对仅扩增 ACE 但不扩增图 9.11 中的其他任何剪接变体。

扩增变体 X 但不扩增变体 Y 的引物对称为 $X+Y$ 对。可以使用 $X+Y$ 对来检测某个在显著剪接变体 Y 背景下的稀有剪接变体 X。然而，关于设计一个寻找全部可变变体的实验和计算规程的问题依旧没有得到解决。

9.8 若干其他的问题和方法

9.8.1 基因预测的隐马氏模型

把 DNA 序列分解成基因的过程可以比喻为把句子解析成文法结构的过程。这个自然的分析比喻被 Searls 和 Dong[312] 进一步推动，他们提倡将语言学的方法用于基因寻找。这种观念在用于基因预测的隐马氏模型中得到了进一步发展[209]，并在程序 GENSCAN 中达到顶点[54]。基因寻找的 HMM 由许多模块构成，每一块识别一个特定的统计特征。例如，剖面隐马氏模型可用来模拟受体位点和供体位点。使用一个不同的 HMM 可以获得密码子的统计量——起始密码子作为初始状态、密码子作为中间状态、终止密码子作为结束状态。在 Burge 和 Karlin[54] GENSCAN 算法中，这些 HMM 被整合在一起。在一个相关方法中，Iseli 等[176] 发展了在 ESTs 中基因预测的 ESTScan 算法。

9.8.2 细菌基因预测

Borodovsky 等[43]第一次使用马尔可夫链对细菌基因进行预测。多种细菌测序计划产生了一个新的计算的复杂问题，即在缺乏任何实验分析数据的情形下如何作计算机基因预测。问题在于缺乏经过实验证实的基因，没有阳性或阴性的测试样本，从而无法从中得到编码和非编码区的统计参数。Frishman 等[113]提出了"similarity-first"方法，它首先在细菌 DNA 中寻找出与数据库中片段紧密相关的片段，并且把它们作为算法的初始序列集。在找到了有相关序列的基因统计参数后，则通过使用一种迭代方式对其他基因进行预测。现在，GenMark[157]，Glimmer[295]和 Orpheus[113]结合了基于相似性的方法和基于统计学的方法。

10 基因组重排

10.1 引言

基因组比较与基因比较 在20世纪80年代末，Jeffrey Palmer和他的同事们在植物细胞器中发现了一种有关进化变异不寻常的新的模式。他们比较了*Brassica oleracea*（甘蓝）和*Brassica campestris*（芜菁甘蓝）的线粒体基因组，两者之间的联系非常紧密（许多基因是99%同源的）。令他们感到惊奇的是，这些分子在基因序列上几乎相同，但在基因排列次序上则有显著性的差别（见图10.1）。这个发现和后来10年中许多其他的研究令人信服地证明了基因组重排是分子进化的一种共同模式。

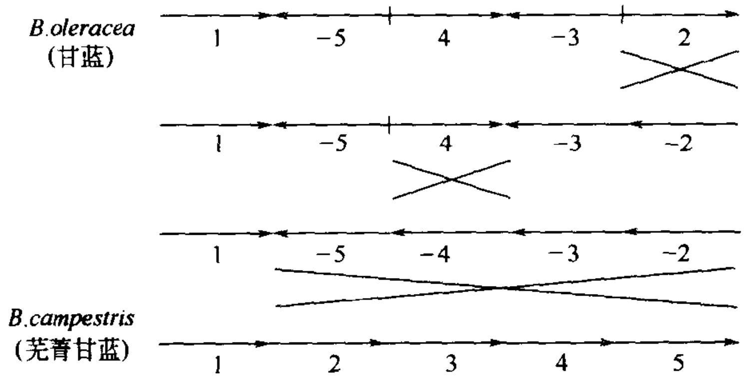


图10.1 从甘蓝基因组到芜菁甘蓝基因组的“变换”

对基因重排的全部研究都包括解决一个组合“难题”——寻找一系列使某一基因组转化成另一个基因组的重排。在图10.1中，你可以看见3个这样的重排把甘蓝“变换”成芜菁甘蓝。如图1.5所示小鼠X号染色体转化成人X号染色体的更复杂的重排方案。所有哺乳动物物种的X号染色体上的高度保守基因（Ohno[255]）提供了一个独立于其他物种的基因组来研究X号染色体进化历史的机会。根据Ohno法则，X号染色体上的基因在哺乳动物最近1.25亿年的发展中几乎没发生改变。然而，X号染色体上的基因次序已经打乱了好几次。

要验证图1.5中6个进化事件所表现的最短连续反序变换——将小鼠X号染色体上的基因次序转化成人X号染色体上的基因次序——并不是很容易。而寻找一个在蛔虫线粒体DNA上的基因次序和人类基因次序之间的最短连续反序变换是个更加困难的计算难题（见图10.2）。

12 31 34 28 26 17 29 4 9 36 18 35 19 1 16 14 32 33 22 15 11 27 5 20 13 30 23 10 6 3 24 21 8 25 2 7

20 5 27 11 15 22 33 32 14 16 1 19 35 18 36 9 4 29 17 26 28 34 31 12 13 30 23 10 6 3 24 21 8 25 2 7

1 16 14 32 33 22 15 11 27 5 20 19 35 18 36 9 4 29 17 26 28 34 31 12 13 30 23 10 6 3 24 21 8 25 2 7

1 16 15 22 33 32 14 11 27 5 20 19 35 18 36 9 4 29 17 26 28 34 31 12 13 30 23 10 6 3 24 21 8 25 2 7

1 16 15 36 18 35 19 20 5 27 11 14 32 33 22 9 4 29 17 26 28 34 31 12 13 30 23 10 6 3 24 21 8 25 2 7

1 16 15 14 11 27 5 20 19 35 18 36 32 33 22 9 4 29 17 26 28 34 31 12 13 30 23 10 6 3 24 21 8 25 2 7

1 16 15 14 31 34 28 26 17 29 4 9 22 33 32 36 18 35 19 20 5 27 11 12 13 30 23 10 6 3 24 21 8 25 2 7

1 26 28 34 31 14 15 16 17 29 4 9 22 33 32 36 18 35 19 20 5 27 11 12 13 30 23 10 6 3 24 21 8 25 2 7

1 26 28 18 36 32 33 22 9 4 29 17 16 15 14 31 34 35 19 20 5 27 11 12 13 30 23 10 6 3 24 21 8 25 2 7

1 26 28 29 4 9 22 33 32 36 18 17 16 15 14 31 34 35 19 20 5 27 11 12 13 30 23 10 6 3 24 21 8 25 2 7

1 26 28 29 30 13 12 11 27 5 20 19 35 34 31 14 15 16 17 18 36 32 33 22 9 4 23 10 6 3 24 21 8 25 2 7

1 26 11 12 13 30 29 28 27 5 20 19 35 34 31 14 15 16 17 18 36 32 33 22 9 4 23 10 6 3 24 21 8 25 2 7

1 26 27 28 29 30 13 12 11 5 20 19 35 34 31 14 15 16 17 18 36 32 33 22 9 4 23 10 6 3 24 21 8 25 2 7

1 26 27 28 29 30 31 34 35 19 20 5 11 12 13 14 15 16 17 18 36 32 33 22 9 4 23 10 6 3 24 21 8 25 2 7

1 26 27 28 29 30 31 34 35 19 20 9 22 33 32 36 18 17 16 15 14 13 12 11 5 4 23 10 6 3 24 21 8 25 2 7

1 26 27 28 29 30 31 22 9 20 19 35 34 33 32 36 18 17 16 15 14 13 12 11 5 4 23 10 6 3 24 21 8 25 2 7

1 26 27 28 29 30 31 32 33 34 35 19 20 9 22 36 18 17 16 15 14 13 12 11 5 4 23 10 6 3 24 21 8 25 2 7

1 26 27 28 29 30 31 32 33 34 35 36 22 9 20 19 18 17 16 15 14 13 12 11 5 4 23 10 6 3 24 21 8 25 2 7

1 26 27 28 29 30 31 32 33 34 35 36 22 9 24 3 6 10 23 4 5 11 12 13 14 15 16 17 18 19 20 21 8 25 2 7

1 26 27 28 29 30 31 32 33 34 35 36 22 9 8 21 20 19 18 17 16 15 14 13 12 11 5 4 23 10 6 3 24 25 2 7

1 26 27 28 29 30 31 32 33 34 35 36 8 9 22 21 20 19 18 17 16 15 14 13 12 11 5 4 23 10 6 3 24 25 2 7

1 26 27 28 29 30 31 32 33 34 35 36 8 9 22 21 20 19 18 17 16 15 14 13 12 11 5 4 3 6 10 23 24 25 2 7

1 26 27 28 29 30 31 32 33 34 35 36 8 9 22 21 20 19 18 17 16 15 14 13 12 11 5 4 3 2 25 24 23 10 6 7

1 2 3 4 5 11 12 13 14 15 16 17 18 19 20 21 22 9 8 36 35 34 33 32 31 30 29 28 27 26 25 24 23 10 6 7

1 2 3 4 5 11 12 13 14 15 16 17 18 19 20 21 22 9 8 7 6 10 23 24 25 26 27 28 29 30 31 32 33 34 35 36

1 2 3 4 5 6 7 8 9 22 21 20 19 18 17 16 15 14 13 12 11 10 23 24 25 26 27 28 29 30 31 32 33 34 35 36

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36

图 10.2 蛔虫线粒体 DNA 转化为人线粒体 DNA 的一个最俭重排（26 个反序）

当基因组只是由少量“保守块”组成的情形，Palmer 和他的合作者们能找到最俭重排方案。但是，对于超过 10 个模块所组成的基因组，在所有的可能解中进行穷举搜索是远远超过“笔头（pen-and-pencil）”方法的能力。其结果，Palmer 和 Herbon[259]，Makaroff 和 Palmer[229]，忽略了在更复杂的例子中的最俭重排方案，例如芜菁甘蓝与黑芥菜相比或者芜菁甘蓝与萝卜相比中的最俭重排方案。

传统的研究分子进化的技术是基因比较，即根据单个基因（或者少数基因）的点突变重建系统树（phylogenetic tree)。在“甘蓝和芜菁甘蓝”例子中，基因比较方法几乎是不适用的，因为在甘蓝和芜菁甘蓝线粒体中基因的点突变比例很低，所以它们的基因几乎是相同的。对于非常缓慢进化的基因组而言，基因组比较（即基因次序的比较）则是一种可供选择的方法。基因组比较比基因比较更具有确定性的另一个进化问题的例子是快速进化的病毒的进化问题。

对疱疹病毒分子进化的研究所提出的问题大大超出他们可以回答的范围。疱疹病毒基因组进化是如此迅速以致现在的表型的极端可能显得相当无关；在疱疹病毒中许多基因之间相似性很低，以致在背景噪声下常常不能被识别出来。因此，经典的序列比较方法对于有高度趋异的基因组不太有效；贸然尝试解决疱疹病毒分子系统发育这个难题可能会引起矛盾，因为不同的基因会产生不同的进化树。疱疹病毒有 70～200 个左右的基因；它们之间共有 7 个保守块，在不同的疱疹病毒基因组中被重排。如图 10.3 所示这些模块在巨细胞病毒（CMV）和 Epstein-Barr 病毒（EBV）中的不同排列，以及将 CMV 基因次序转化成 EBV 基因次序的一个最短连续反序变换（Hannenhalli 等[152]）。这种大尺度重排（5 个反序）的次数比起 CMV 和 EBV 之间点突变的数目（几十万）是非常少的。因此，在基因组水平上对这种重排进行分析可以弥补传统的基因水平上的分子进化研究的不足。基因组比较与经典的基因比较相比具有特定的优点和缺点：基因组比较忽视基因的真实 DNA 序列，而基因比较则忽视了基因的次序。最终的目标应是将基因组比较和基因比较的优点结合到单一的算法中。

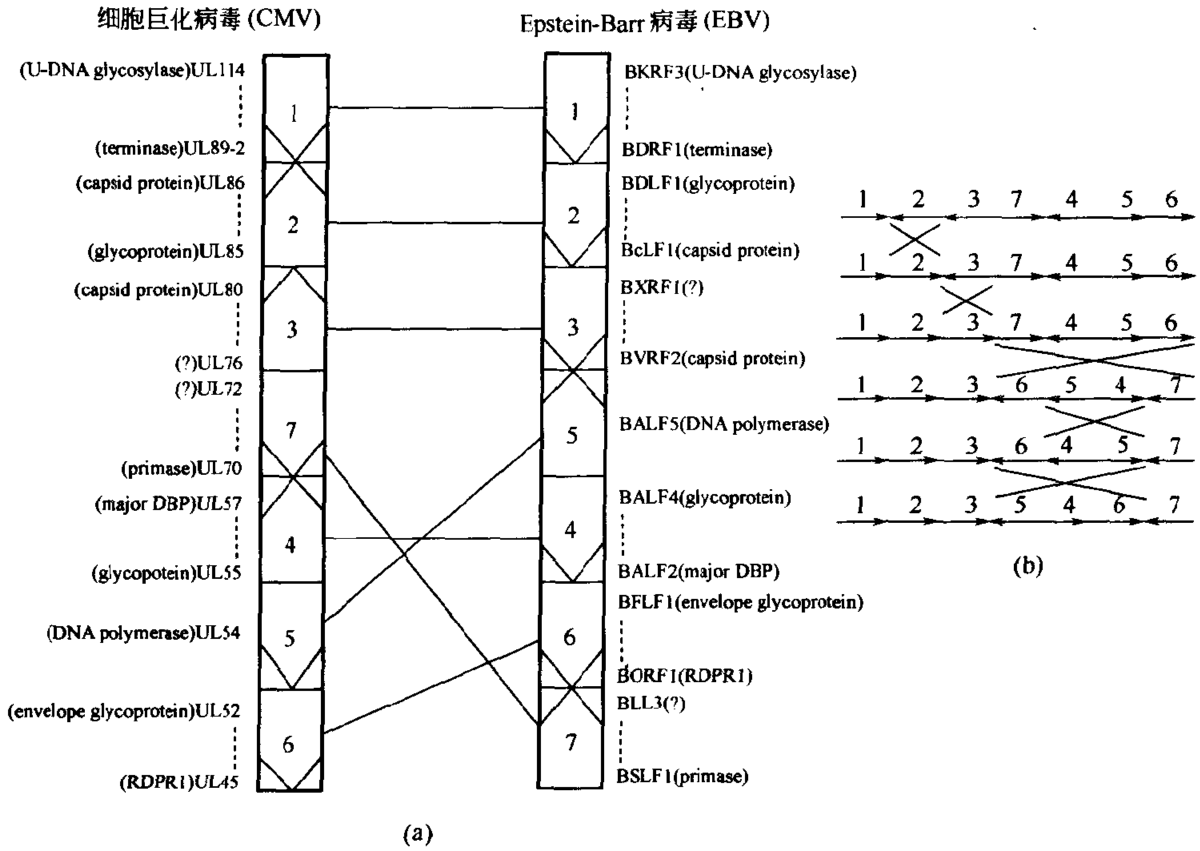


图 10.3 比较基因组组织和 CMV 基因次序转化为 EBV 基因次序的最短连续重排

在 20 世纪 30 年代末，Dobzhansky 和 Sturtevant 成为在分子生物学领域中对基因组重排进行分析的先驱，他们发表了一篇里程碑似的论文——关于果蝇物

种 17 个倒位的重排方案（Dobzhansky 和 Sturtevant[87]）。随着大规模作图和测序的出现，在不同的领域内有关基因组比较问题的数目迅速增长，这些领域包括病毒、细菌、酵母、植物和动物的进化。

反序排序法 David Sankoff 首次提出了基于基因次序比较的计算方法（Sankoff 等[302,304]和 Sankoff[300]）。基因组重排可模型化为一个反序排序法的组合问题，其描述如下。两个生物体中基因的次序记为排列 $\pi=\pi_1\pi_2\cdots\pi_n$ 和 $\sigma=\sigma_1\sigma_2\cdots\sigma_n$。区间 $[i, j]$ 的一个反序 $\rho(i, j)$ 记为排列

$$\begin{pmatrix} 1 & 2\cdots i-1 & i & i+1\cdots j-1 & j & j+1\cdots n \\ 1 & 2\cdots i-1 & j & j-1\cdots i+1 & i & j+1\cdots n \end{pmatrix}$$

显然，$\rho(i, j)$ 具有将 $\pi_i\pi_{i+1}\cdots\pi_j$ 倒换的效果，并且将 $\pi_1\cdots\pi_{i-1}\pi_i\cdots\pi_j\pi_{j+1}\cdots\pi_n$ 变换成 $\pi\rho(i, j)=\pi_1\cdots\pi_{i-1}\pi_j\cdots\pi_i\pi_{j+1}\cdots\pi_n$。

给定排列 π 和 σ，反序距离问题就是寻找一系列反序 $\rho_1, \rho_2, \cdots, \rho_t$，使得 $\pi\rho_1\rho_2\cdots\rho_t=\sigma$，且 t 为最小。我们称 t 是 π 和 σ 之间的反序距离。按反序排序 π 就成为判定 π 和恒等排列（$12\cdots n$）之间反序距离 $d(\pi)$ 的问题。

计算机科学家们已经研究了相关的前缀反序排序（sorting by prefix reversal）问题，也称为磁带翻转问题（pancake flipping problem）：给定一个任意的排列 π，寻找 $d_{\text{pref}}(\pi)$，是对 π 进行排序时形如 $\rho(1, i)$ 的反序的最小数目。Harry Dweigter 通过对“现实生活”情形的描述产生了磁带翻转问题。

The chef in our place is sloppy, and when he prepares a stack of pancakes they come out all different sizes。Therefore, when I deliver them to a customer, on the way to a table I rearrange them (so that the smallest winds up on top, and so on, down to the largest at the bottom) by grabbing several from the top and flipping them over, repeating this (varying the number I flip) as many times as necessary。If there are n pancakes, what is maximum number of flips that I will ever have to use to rearrange them?

Bill Gates（一个 20 世纪 70 年代末 Harvard 没毕业的学生，现在在微软）和 Cristos Papadimitriou 第一次尝试解决这个问题（Gates 和 Papadimitriou[120]）。他们证明了：对称群的前缀反序直径（prefix reversal diameter）$d_{\text{pref}}(n)=\max_{\pi\in S_n} d_{\text{pref}}(\pi)$，小于等于 $\frac{5}{3}n+\frac{5}{3}$，并且当 n 充分大时 $d_{\text{pref}}(n)\geq\frac{17}{16}n$。磁带翻转问题还没有得到解决。

断点图 是什么使得排序一个排列变得困难？在基因组重排的早期计算研究中，Watterson 等[366]以及 Nadeau 和 Taylor[248]分别引进了断点的概念，并且注意到一些反序距离与断点数之间的相互关系。（事实上，早在 60 年前，Sturtevant 和 Dobzhansky[331]已经隐含地讨论了这些相互关系！）下面给出断点的定义。

如果 $|i-j|=1$，记 $i\sim j$。通过增加 $\pi_0=0$ 和 $\pi_{n+1}=n+1$ 来扩展排列 $\pi=\pi_1\pi_2\cdots\pi_n$。如果 $\pi_i\sim\pi_{i+1}$，称元素对 $(\pi_i,\ \pi_{i+1})$，$0\leqslant i\leqslant n$，是 π 的一个邻接；如果 $\pi_i\nsim\pi_{i+1}$，则称为断点（见图 10.4）。恒等排列没有断点，而反序排序则对应于消除断点。注意到每个反序最多能够消除两个断点，因此立即可得 $d(\pi)\geqslant\frac{b(\pi)}{2}$，其中 $b(\pi)$ 是 π 中的断点数。基于断点的概念，Kececioglu 和 Sankoff[194] 发现了一个具有性能保证为 2 的反序排序的近似算法。他们还推导了有效范围，当 n 在 30～50 范围内改变时几乎最优地解决反序距离问题。这个范围覆盖了动物线粒体基因组所有生物学上的重要情形。

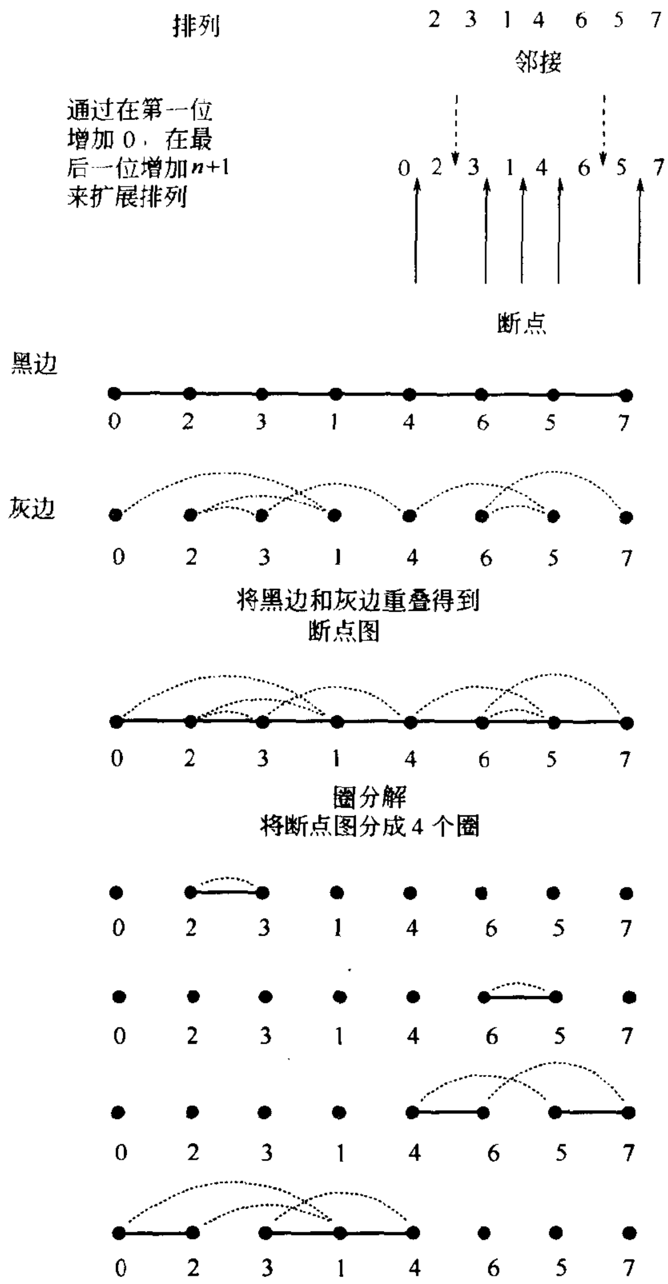


图 10.4 断点、断点图和最大圈分解

然而，用断点对反序距离进行估计是非常不精确的。Bafna 和 Pevzner[19] 指出另一个参数（断点图的最大圈分解的长度）估计反序距离具有较大的精确性。

排列 π 的断点图是一个有 $n+2$ 个顶点 $\{\pi_0,\ \pi_1,\ \cdots,\ \pi_n,\ \pi_{n+1}\}\equiv\{0,\ 1,\ \cdots,\ n,\ n+1\}$ 的边染色图 $G(\pi)$。对 $0\leqslant i\leqslant n$，使用黑边连接顶点 π_i 和 π_{i+1}。如果 $\pi_i\sim\pi_j$，则用灰边连接顶点 π_i 和 π_j。如图 10.4 所示，将按照排列 π 给定的次序穿过顶点 0，1，…，n，$n+1$ 的黑色路径和灰色路径进行重叠，得到的一张断点图。

如果边染色图 G 中的某个圈里每两个连续边的颜色是截然不同的，那么这个圈称为是交错的。下文中的圈就是指交错圈。图 G 中的一个顶点 v 称为平衡的，如果进入 v 的黑边数等于进入 v 的灰边数。平衡图中的每一个顶点都是平衡的。显然，$G(\pi)$ 是平衡图，所以它包含一个交错欧拉圈。因此，在边不相连

的交错圈中存在一个 G (π) 的圈分解（平衡图中的每一边恰好属于圈分解中的某一圈)。边分解中的圈可能是自相交的。图 10.4 中的断点图可以被分解成 4 个圈，其中一个是自相交的。令人感兴趣的是将断点图分解成具有最大数目$c(\pi)$的边不相连的交错圈。对于图 10.4 中的排列，$c(\pi)=4$。

在对反序距离进行估计时，圈分解扮演一个很重要的角色。当对排列进行反序时，最大分解中的圈的个数最多可以被改变一次（而断点数可以被改变两次)。Bafna 和 Pevzner[19]证明了 $d(\pi)\geqslant n+1-c(\pi)$，比在断点情况下的 $d(\pi)\geqslant\frac{b(\pi)}{2}$ 更严格。对大多数生物学例子，$d(\pi)=n+1-c(\pi)$，于是把反序距离问题归结为最大圈分解问题。

有符号排列的对偶定理 寻找最大圈分解是一个困难问题。幸好，在和生物学更有关的有符号排列情形中，这个问题是普通的。基因是 DNA 的有向片段，基因组中一个包含 n 个基因的序列可以用 $\{1, \cdots n\}$ 上的有符号的排列来表示，这个排列带有和 π 中每一个元素相关的＋或－号。例如，在图 10.1 中甘蓝的基因次序可表示为有符号排列（+1 －5 +4 －3 +2)。在有符号情形下，片段 $[i, j]$ 的每一个反序不但改变次序而且还改变片段中元素的符号（见图 10.1)。感兴趣的是把有符号排列 π 变换成单位有符号排列（+1 +2…+n）所要求的最小反序个数 $d(\pi)$。

Bafna 和 Pevzner[19]注意到断点图的概念可以自然地扩充到有符号排列，只要用无向元素 i_a 和 i_b 替代有向元素 i 的尾部和头部来模拟每一个有向元素 i 即可（见图 10.5)。对有符号排列，限制 $d(\pi)\geqslant n+1-c(\pi)$ 无论对于模拟数据还是对于生物学数据都非常好地近似反序距离。这个最引人兴趣的特性提出了一个问题：这个限制 $d(\pi)\geqslant n+1-c(\pi)$ 是否忽视了其他能缩小 $d(\pi)$ 和 $n+1-c(\pi)$ 之间的间隙的参数（除了最大圈分解的大小)。Hannenhalli 和 Pevzner[154]揭示了另一个“隐藏”的参数（π 中障碍数）使得对有符号排列的排序变得困难，同时指出

$$n+1-c(\pi)+h(\pi)\leqslant d(\pi)\leqslant n+2-c(\pi)+h(\pi) \tag{10.1}$$

其中 h (π) 是 π 中障碍数目。他们还证明了有符号排列的对偶定理，并设计了计算 d (π) 的多项式算法。

无符号排列和比较物理作图 由于按反序排序（无符号的）排列是 NP 难题 (Caprara[57])，许多研究者设法设计一个按反序排序（无符号排列）的实用的近似算法。

π 的一个块是一个不含断点的区间 $\pi_i\cdots\pi_j$，即对于 $0\leqslant i\leqslant k<j\leqslant n+1$，($\pi_k$, π_{k+1}) 是一个邻接。把 π 的一个带定义为最大的块，即满足 (π_{i-1}, π_i) 和 (π_j, π_{j+1}) 是块 $\pi_i\cdots\pi_j$ 的断点。一个元素的带称为单元素 (singleton)，两个元素的

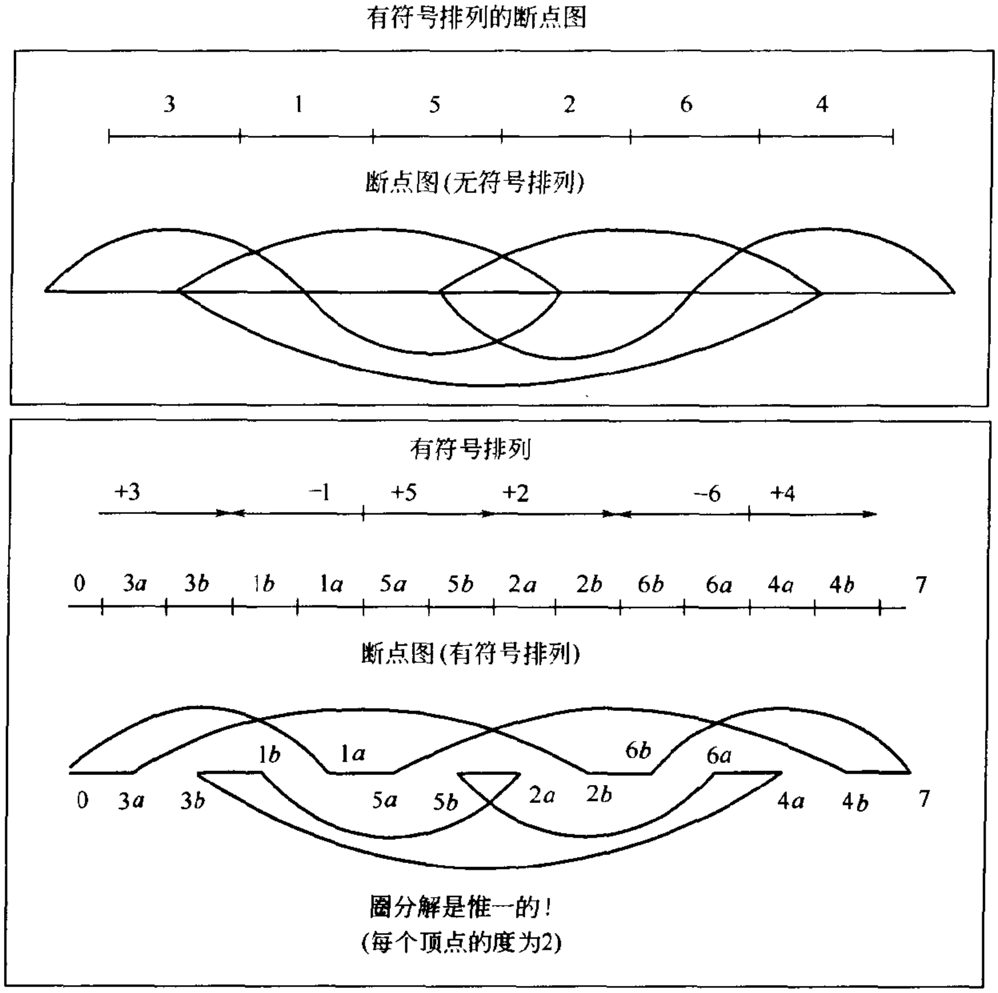


图 10.5 用一个无符号排列来模拟一个有符号排列

带称为 2-带（2-strip），拥有多于 2 个元素的带称为长带。由此可见单元素是按反序排序无符号排列的主要难点。

如果 $k<i\leqslant l$ 或者 $k\leqslant j<l$，则反序 $\rho(i, j)$ 切割带 $\pi_k\cdots\pi_l$。一个切割带的反序将那些在恒等排列中连续的元素分开。因此非常自然地想到，对每一个排列 π，存在一个 π 的按反序（最佳的）排序，使得其不切割任何带。然而，这是错误的。如果不切割带的话，排列 3412 需要 3 个反序，但它通过两步就可被排序：3412→1432→1234。Kececioglu 和 Sankoff[192] 推测每个排列都有一个按反序的最佳排序，且不切割长带又不增加断点数目。由于恒等排列没有断点，而按反序排序对应于消除断点。从这个观点出发，自然期望对每个排列存在一个按反序的最佳排序且又不增加断点数。Hannenhalli 和 Pevzner[155] 利用有符号排列的对偶定理证明了“反序不切割长带”和“反序不增加断点数”这两个猜想。

生物学家通过对整个基因组测序或使用比较物理图谱得到基因次序。测序提供有关基因方向的信息并且允许通过有符号的排列来表示基因组。但是，对整个

基因组进行测序的花费依然昂贵，现在能得到的有关基因次序的实验数据大部分是基于比较物理图谱的。物理图谱常常不提供有关基因方向的信息，因此导致用无符号排列 π 来表示基因组。生物学家试图通过这种表示方式得到一个有符号排列，即对 π 中增加（减少）的带赋予正（负）号（见图 10.6）。“反序不切割长带”这个性质给出了在长带情况下这个过程的理论证明。同时，对于 2-带，这种程序或许不能找到最佳重排方案。Hannenhalli 和 Pevzner[155] 指出了一个使该过程无效的生物学例子，同时还给出了一个解决这个问题的算法。

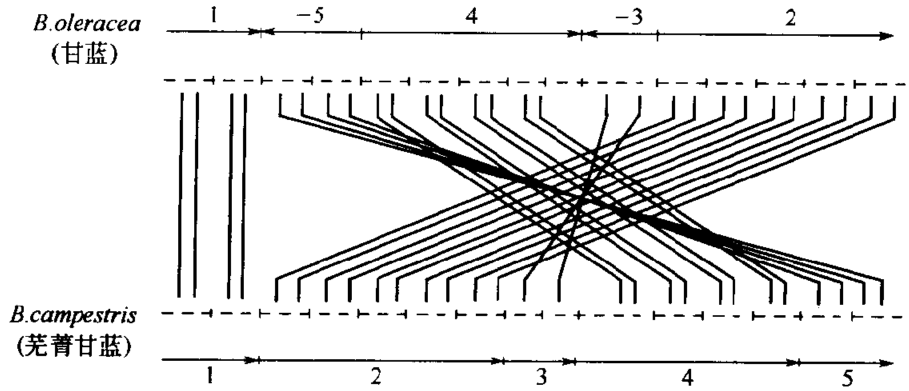


图 10.6 甘蓝和芜菁甘蓝的比较物理图谱（无符号排列）和相应的有符号排列

没有单元素的排列称为无单元素（singleton-free）排列。分析这种排列的困难在于选择“是否切割”2-带。“切割”一组 2-带的特征（Hannenhalli 和 Pevzner[155]，产生了排序无单元素排列的多项式算法和排序有少量单元素排列的多项式算法。这些算法可以适用于分析来自比较物理图谱的重排方案。

低分辨率的物理图谱通常包含许多单元素，所以这种图谱的重排方案是难以分析的。如果单元素的数目是 $O(\lg n)$，那么 Hannenhalli 和 Pevzner[155] 的算法耗时为多项式时间。这表明比较物理图谱的分辨率在分子进化研究中希望能达到 $O(\lg n)$ 个单元素。这就建议在分子进化研究中对于比较物理作图而言，$O(\lg n)$ 个单元素是分辨率权衡所要求的。如果单元素的数目过大，生物学家交错选择附加实验（即对一些区域测序）来确定基因方向。

多染色体基因组的重排 当格林兄弟在“Puss in Boots”中讲述人变成小老鼠的童话故事的时候，他们几乎没有预期到两个世纪后人类和小鼠会成为遗传研究最多的哺乳动物。人-小鼠比较物理作图开始于 20 年前，现在，这两个种类中的上千对同源基因已经完成作图。结果，生物学家已经发现了人和小鼠之间相关基因不是无秩序地分布在基因组上，而是形成“保守块（conserved blocks)”。目前的比较作图数据指出，人基因组和小鼠基因组都包含有约 150 个块，这些块在两者之间有所“移动”（Copeland 等[74]）。例如，人的第 7 号染色体可以被看作为在小鼠的第 2，5，6，11，12，13 号染色体上的不同基因的镶嵌物（见图

1.4)。块的移动发生相当少（每百万年大概发生一次），这就为生物学家重构有关人-小鼠进化的重排方案带来了希望。Nadeau 和 Taylor[248] 在他们开创性的论文中估计，自从八千万年前人和小鼠发生进化分叉后，两者之间令人惊讶地只发生了极少的基因重排（178±79）。

在考虑的模型中，每一个基因被表示成一个整数，（“+”或“−”）号表示基因的方向。染色体被定义为基因的序列，而基因组则定义为一套染色体。给定两个基因组 Π 和 Γ，感兴趣的是 Π 演化成 Γ 的一个最俭方案，即对于要求把 Π 变换成 Γ 的重排事件（在下面定义）的最短序列。下面假定 Π 和 Γ 包含相同的基因集合。如图 10.7 所示一个基因组变换为另一个基因组的 4 个重排事件。

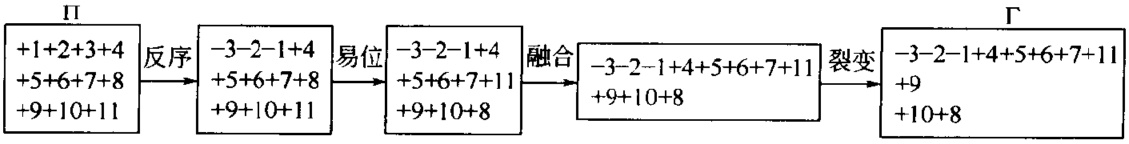


图 10.7　基因组 Π 进化成基因组 Γ

设 $\Pi=\{\pi(1),\ \cdots,\ \pi(N)\}$ 是由 N 条染色体组成的基因组，且 $\pi(i)=\{\pi(i)_1,\ \cdots,\ \pi(i)_{n_i}\}$，其中 n_i 是第 i 条染色体上的基因的数目。每一条染色体 π 可看作是“从左到右”的［即 $\pi=(\pi_1\cdots\pi_n)$］也可以看作是“从右到左”的［即 $-\pi=(-\pi_n\cdots-\pi_1)$］，这两种观察导致对同一条染色体的两个等价的表示（即染色体的方向是没关系的）。在多染色体基因组中四个最常见的基本重排事件是反序、易位、融合和裂变，下面给出它们的定义。

设 $\pi=\pi_1\cdots\pi_n$ 是一条染色体，$1\leqslant i\leqslant j\leqslant n$。一条染色体 π 上的一个反序 $\rho(\pi,\ i,\ j)$ 将 $\pi=\pi_1\cdots\pi_{i-1}\pi_i\cdots\pi_j\pi_{j+1}\cdots\pi_n$ 内的基因重新排列，并把 π 变换为 $\pi_1\cdots\pi_{i-1}-\pi_j\cdots-\pi_i\pi_{j+1}\cdots\pi_n$。设 $\pi=\pi_1\cdots\pi_n$，$\sigma=\sigma_1\cdots\sigma_m$ 是两条染色体，$1\leqslant i\leqslant n+1$，$1\leqslant j\leqslant m+1$。易位 $\rho(\pi,\ \sigma,\ i,\ j)$ 调换染色体 π 和 σ 之间的基因，将两条染色体变换为 $\pi_1\cdots\pi_{i-1}\sigma_j\cdots\sigma_m$ 和 $\sigma_1\cdots\sigma_{j-1}\pi_i\cdots\pi_n$，它们分别含有 $(i-1)+(m-j+1)$ 个基因和 $(j-1)+(n-i+1)$ 个基因。我们记基因组 Π 经过重排（反序或易位）ρ 后得到的基因组为 $\Pi\cdot\rho$。给定基因组 Π 和 Γ，基因组排序问题就是寻找一系列反序和易位 $\rho_1\cdots\rho_t$，使得 $\Pi\cdot\rho_1\cdots\rho_t=\Gamma$ 并且 t 是最小的。我们称 t 为 Π 与 Γ 之间的基因组距离（genomic distance），基因组距离问题即寻找 Π 与 Γ 之间的基因组距离 $d(\Pi,\ \Gamma)$。

易位 $\rho(\pi,\ \sigma,\ n+1,\ 1)$ 连接染色体 π 和 σ，结果生成染色体 $\pi_1\cdots\pi_n\sigma_1\cdots\sigma_m$ 和一个空染色体 $\varnothing$。这种特殊的易位，引起（非空）染色体数目的减少，这在分子生物学中称为融合。易位 $\rho(\pi,\ \varnothing,\ i,\ 1)$，$1<i<n$，把染色体 π“破裂”成两条染色体 $(\pi_1\cdots\pi_{i-1})$ 和 $(\pi_i\cdots\pi_n)$。这种易位，引起（非空）染色体数目增加，称为裂变。在哺乳动物进化过程中，融合和裂变是相当普通的：例如，人类

和黑猩猩全部基因组组织中的主要差别在于黑猩猩的两条染色体融合成人类的一条染色体。

Kececioglu 和 Ravi[191]第一次尝试分析多染色体基因组的重排。他们的近似算法解决了包含相同染色体数目时两个基因组的重排。这种算法存在严重的局限性，因为不同的生物体（特别是人和小鼠）拥有不同的染色体数。从这个观点出发，每一个现实的基因组重排模型应该包括融合和裂变。这表明，在分析基因组重排中融合和裂变是主要的困难。Hannenhalli 和 Pevzner[153]证明了多染色体基因组的对偶定理，根据反映字符串的集合不同组合性质的 7 个参数计算基因距离。基于这个结果，他们发现了解决这个问题的多项式算法。

这种分析方法的思想是把 Π 和 Γ 中的 N 个染色体分别连接成排列 π 和 γ，通过按反序对 π 和 γ 排序来模拟对 Π 和 Γ 的基因组排序。这个方法的困难在于，对于 Π 和 Γ 有 $N!2^N$ 个不同的连接方法，其中仅有一些是最佳连接，模拟 Π 和 Γ 之间的最佳排序。Hannenhalli 和 Pevzner[153]引进了染色体的翻转（flipping）和加帽（capping）技术，使人们可找到一个最佳连接。

当然，仅仅两个基因组的基因次序还不足以描绘出一个恰当的重排方案。比较基因作图使得许多哺乳动物物种比较图谱的产生成为可能（O’ Brien 和 Graves[254]）。但是，这些图谱的分辨率比人-小鼠图谱的分辨率明显低。由于比较物理作图是非常费力的工作，因此几乎不能期待会在其他哺乳动物基因组中重现在人-小鼠图谱得到的极大成就。然而，一种称为染色体着色（chromosome painting）的实验技术使研究人员不必实际构建一个精确的“基于基因的”图谱就能得到基因次序。在过去，染色体着色技术的应用仅限于灵长类的动物（Jauch 等[178]）；把这种方法应用到其他哺乳动物的尝试并不成功，因为在远源物种之间的 DNA 序列存在多样性。后来，Scherthan 等[307]发展了染色体着色的改进方法，称为 ZOO-FISH，能够在远亲哺乳动物种中发现同源染色体片段。使用 ZOO-FISH，Rettenberger 等[284]很快地完成了人-猪的染色体着色计划，同时识别出 47 个在人和猪中共有的保守块。人-猪的染色体着色计划的成功预示使用 ZOO-FISH 可以廉价地得到许多哺乳动物的基因次序，这样就为解决已有 100 年历史的有关哺乳动物进化的问题，提供了一个无价的新的数据源。

10.2 断点图

圈分解是相当奇怪的概念，起初它会被认为和基因组重排有一点相似。但是，观察到一次反序将最大分解中的圈个数最多改变 1，这允许根据最大圈分解限制反序距离。

定理 10.1 对每一个排列 π 和反序 ρ，有 $c(\pi\rho)-c(\pi)\leqslant 1$。

证明 一个任意的反序 $\rho(i, j)$ 包含图 $G(\pi)$ 中的 4 个顶点，且用黑边

$ADD=\{(\pi_{i-1}, \pi_j), (\pi_i, \pi_{j+1})\}$ 替换两条黑边 $DEL=\{(\pi_{i-1}, \pi_i), (\pi_j, \pi_{j+1})\}$。

如果 ADD 中的这两条黑边同属于最大圈分解 $G(\pi\rho)$ 里的一个圈，那么删除该圈就会产生一个至少有 $c(\pi\rho)-1$ 个圈的 $G(\pi)$ 圈分解。因此，$c(\pi)\geqslant c(\pi\rho)-1$。

另一方面，如果 ADD 中的黑边属于 $G(\pi\rho)$ 的最大圈分解中的不同的圈 C_1 和 C_2，那么删除 $C_1\cup C_2$ 在图 $G(\pi\rho)\setminus(C_1\cup C_2)$ 中给出大小为 $c(\pi\rho)-2$ 的一组边不相交圈。显然，边的集合 $(C_1\cup C_2\cup DEL)\setminus ADD$ 形成一个平衡图，且必含有至少一个圈。结合前面得到的 $c(\pi\rho)-2$ 个圈，得到 $G(\pi)=(G(\pi\rho)\setminus(C_1\cup C_2))\cup(C_1\cup C_2\cup DEL\setminus ADD)$ 的一个圈分解，它至少有 $c(\pi\rho)-1$ 个圈。 ■

综合定理 10.1 和对恒等排列 l 成立的 $c(l)=n+1$，立即可以得出 $d(\pi)\geqslant c(l)-c(\pi)\equiv n+1-c(\pi)$。

定理 10.2 对每一个排列 π，$d(\pi)\geqslant n+1-c(\pi)$。

10.3 难以排序的排列

定义 $d(n)=\max\limits_{\pi\in S_n} d(\pi)$ 是一个 n 阶对称群的反序直径。Gollan 推测 $d(n)=n-1$，并且仅有一个排列 γ_n 和它的反向排列 γ_n^{-1} 需要 $n-1$ 次反序进行排序。用单行记号，Gollan 排列可定义如下

$$\gamma_n=\begin{cases}(3,1,5,2,7,4,\cdots,n-3,n-5,n-1,n-4,n,n-2), & n\text{ 为奇数}\\(3,1,5,2,7,4,\cdots,n-6,n-2,n-5,n,n-3,n-1), & n\text{ 为偶数}\end{cases}$$

Bafna 和 Pevzner[19] 通过证明 $c(\gamma_n)=2$ 和应用定理 10.2 证明了 Gollan 的猜想。而且，他们证明两个随机排列之间的反序距离非常接近于对称群的反序直径，从而指出在分子进化研究中反序距离提供了相关序列和无关序列之间一个好的分离。

我们证明断点图 $G(\gamma_n)$ 最多有两个不相交的交错圈。由顶点 $\{4, 5, \cdots, n-5, n-4\}$ 形成 $G(\gamma_n)$ 的一个子图具有正则结构（见图 10.8）。在图 10.8 的子图中任意一个圈的黑边的方向是从底部到顶部，所有灰边的方向则是从顶部到底部。注意在这定位下，所有边被指向↖，或↙，或↓，因此沿着这个圈的边走最终可以缓慢地到达左边。如何回到最初的顶点？只有在到达任一个起"转向点"作用的"非正则"顶点（1 和 3）以后才可以这样做。接下来的引理证明这个启发式的结论是对的。

引理 10.1 $G(\gamma_n)$ 中的每一个交错圈包含顶点 1 或 3。

证明 设 i 是 $G(\gamma_n)$ 中的一个交错圈 X 的最小奇数顶点。考虑 X 中连续顶点序列 i，j，k，其中 (i, j) 是黑的，(j, k) 是灰的。如果 $i>5$，则 $j=i-3$

或者 $j=i-5$ 且 $k=j+1$ 或者 $k=j-1$（见图 10.8），暗示 k 是奇数，且 $k<i$，矛盾。如果 $i=5$，则 $j=2$，且 $k=1$ 或 $k=3$，矛盾。因此，i 是 1 或 3。 ■

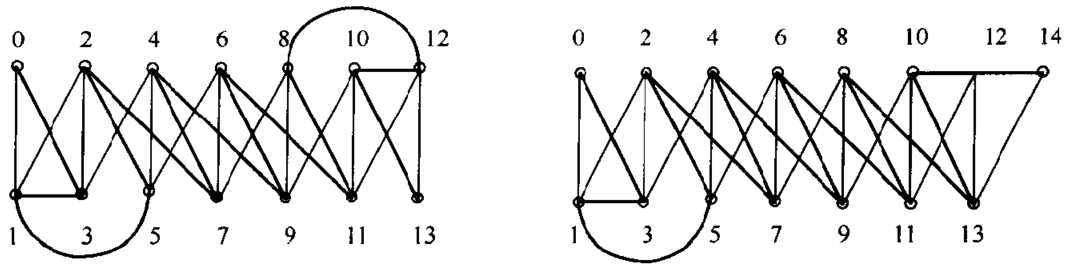


图 10.8 $G(\gamma_{12})$ 和 $G(\gamma_{13})$

定理 10.3 （Gollan 猜想）对每一个 n，$d(\gamma_n)=d(\gamma_n^{-1})=n-1$。

证明 当 $n\leqslant 2$ 时，证明显而易见。当 $n>2$ 时，将 $G(\gamma_n)$ 的顶点集分割成 $V_l=\{0,1,3\}$ 和 V_r。由引理 10.1 以及 V_l 中不包含圈的事实，可知每一个交错圈必包含至少两条来自切割 (V_l, V_r) 的边。因为切割 (V_l, V_r) 由 4 条边 $(1,2)$，$(1,5)$，$(3,2)$，$(3,4)$ 组成，所以在 $G(\gamma_n)$ 的一个圈分解中的边不相交的交错圈的最大个数不超过 $\frac{4}{2}=2$。

从定理 10.2 可得，$d(\gamma_n)\geqslant n+1-c(\gamma_n)\geqslant n-1$。另一方面，$d(\gamma_n)\leqslant n-1$，因此存在一个简单的算法，在 $n-1$ 步中完成对每个含 n 个元素的排列进行排序。最后注意到 $d(\gamma_n^{-1})=d(\gamma_n)$。 ■

Bafna 和 Pevzner[19] 还证明了 γ_n 和 γ_n^{-1} 是 S_n 中惟一的反序距离为 $n-1$ 的排列。

定理 10.4 （强 Gollan 猜想）对每一个 n，γ_n 和 γ_n^{-1} 是惟一要求 $n-1$ 个反序进行排序的排列。

10.4 期望的反序距离

对任意一个排列 $\pi\in S_n$，考虑形成最大分解的一组圈，并依照大小对它们进行分割。记 $c_i(\pi)$ 是在最大分解中长度为 i 且不包含顶点 0 或 $n+1$ 的交错圈的数目。记 $\delta\leqslant 2$ 是在最大分解中包含顶点 0 或 $n+1$ 的交错圈的数目。那么

$$c(\pi)=\sum_{i=2}^{2(n+1)} c_i(\pi)+\delta \tag{10.2}$$

对 $k\leqslant 2(n+1)$，考虑在分解中大小至少是 k 的圈。这种圈的数目是 $c(\pi)-\sum_{i=2}^{k-1} c_i(\pi)-\delta$。现在，$\pi$ 的断点图严格地有 $2(n+1)$ 条边。综合这一点和这个圈是边不相交的，有

$$\forall k\leqslant 2(n+1),\ c(\pi)-\sum_{i=2}^{k-1} c_i(\pi)-\delta\leqslant\frac{1}{k}\left[2(n+1)-\sum_{i=2}^{k-1} ic_i(\pi)\right] \tag{10.3}$$

和

$$\forall k \leqslant 2(n+1),\ c(\pi) \leqslant \frac{1}{k}\left[2(n+1)+\sum_{i=2}^{k-1}(k-i)c_i(\pi)\right]+\delta \qquad (10.4)$$

由定理 10.2，不等式（10.4）和 $\delta \leqslant 2$ 可以推出对所有的 $k \leqslant 2(n+1)$，可以限制 $d(\pi)$ 为

$$d(\pi) \geqslant \left(1-\frac{2}{k}\right)(n+1)-\frac{1}{k}\left[\sum_{i=2}^{k-1}(k-i)c_i(\pi)\right]-2 \qquad (10.5)$$

$$d(\pi) \geqslant \left(1-\frac{2}{k}\right)(n+1)-\left[\sum_{i=2}^{k-1}c_i(\pi)\right]-2 \qquad (10.6)$$

考虑一随机均匀生成的排列 π。在 $G(\pi)$ 的最大圈分解中长度为 i 的圈的期望数目记为 $E[c_i(\pi)] = \frac{1}{n!}\sum_{\pi \in S_n} c_i(\pi)$。如果能限制 $E[c_i(\pi)]$，可以用式（10.6）给出期望反序距离的一个下界。引理 10.2 提供这样一个范围，有点令人惊讶的是它不依赖于 n。注意，$E[c_i(\pi)]$ 的定义中有一个稍微不明确的地方，对每个 $\pi \in S_n$，它依赖于最大圈分解的选取。然后，这不影响引理 10.2 对任意的圈分解成立。

引理 10.2　$E[c_i(\pi)] \leqslant \frac{2^i}{i}$。

证明　长度为 $i=2t$ 的圈包含 t 条黑边（顶点的无序对），形如

$$\{(x'_t, x_1), (x'_1, x_2), (x'_2, x_3), \cdots, (x'_{t-1}, x_t)\}, x_j \sim x'_j$$

考虑集合 $x_1, x_2, \cdots, x_t$。首先，要求在每个最大圈分解中 $x_1, x_2, \cdots, x_t$ 是各不相同的。这是因为，若对某些 $1 \leqslant k < l \leqslant t$，考虑 $x_k = x_l$ 情况，则 (x'_k, x_{k+1})，(x'_{k+1}, x_{k+2})，…，$(x'_{l-1}, x_l = x_k)$ 形成一个交错圈，该圈可被分开以得到更大的分解。

有 $\frac{n!}{(n-t)!}$ 种方法选择有序集 $x_1, x_2, \cdots, x_t$。一旦选定有序集后，对每个 x'_j 最多有两个元素的选择，则它限定长度为 $2t$ 的圈数目不超过 $2^t\frac{n!}{(n-t)!}$。注意计数每个 $(2t)$-圈 $2t$ 次，因此长度为 $2t$ 的圈的数目更严格的上限是 $\frac{2^t}{2t}\frac{n!}{(n-t)!}$。

选择一个任意的 $(2t)$-圈。这个圈能产生的排列数目不超过交换剩下的 $n-2t$ 元素的方法个数加上形成该圈的 t 对的数目。另外，可以翻转每一对得到不同的次序，这最多可以给出 $2^t(n-t)!$ 个排列。设 p 是任意一个 $(2t)$-圈在一个随机排列中出现的概率。那么 $p \leqslant \frac{2^t(n-t)!}{n!}$ 并且

$$E(c_i(\pi))=E(c_{2t}(\pi))\leqslant \sum_{\{\text{所有}(2t)\text{-圈}\}} p\leqslant \frac{2^{2t}}{2t}=\frac{2^i}{i}$$ ■

长度 2 的圈对应排列 π 中的邻接，共有 $2n$ 个有序邻接。任何这种对肯定能在 $(n-1)!$ 个排列中出现，因此它在随机排列中发生的概率是$\frac{1}{n}$。于是，邻接的期望数是$\frac{2n}{n}$且 $E(c_2)=2$。注意到一个随机排列中断点的期望数是 $n-1$。

使用引理 10.2 和 $E(c_2)=2$ 得到期望反序直径的界限。

定理 10.5　(Bafna 和 Pevzner[19]) $E[d(\pi)]\geqslant\left(1-\frac{4.5}{\lg n}\right)n$。

证明　根据不等式 (10.6)，对所有 $k\leqslant 2(n+1)$

$$E(d(\pi))\geqslant\left(1-\frac{2}{k}\right)(n+1)-\sum_{i=2}^{k-1}E(c_i)-2\geqslant\left(1-\frac{2}{k}\right)(n+1)-\sum_{i=4}^{k-1}\frac{2^i}{i}-4\geqslant n-\frac{2n}{k}-2^k$$

取 $k=\lg\frac{n}{\lg n}$。那么 $2^k\leqslant\frac{n}{k}$，并且当 $n\geqslant 2^{16}$ 时，

$$E(d(\pi))\geqslant\left(1-\frac{3}{\lg\frac{n}{\lg n}}\right)n\geqslant\left(1-\frac{4.5}{\lg n}\right)n$$

$k=10$ 时，由引理 10.2 和不等式 (10.5) 可知当 $19<n<2^{16}$ 时，$E(d(\pi))\geqslant\left(1-\frac{4.5}{\lg n}\right)n$。当 $1\leqslant n\leqslant 19$ 时，$\left(1-\frac{4.5}{\lg n}\right)n<1$。 ■

10.5　有符号排列

设 $\vec{\pi}$ 是 $\{1,\cdots,n\}$ 的有符号排列，即每个元素都有＋号或－号的排列。从一个 n 阶有符号排列 $\vec{\pi}$ 到一个 $\{1,\cdots,2n\}$ 的（无符号）排列 π 的变换定义如下。为了模拟 $\vec{\pi}$ 中元素的符号，用 $2x-1$，$2x$ 替换正元素 $+x$，用 $2x$，$2x-1$ 替换负元素 $-x$ [见图 10.9 (c)]。称无符号排列 π 是有符号排列 $\vec{\pi}$ 的像。在断点图 $G(\pi)$ 中，元素 $2x-1$ 和 $2x$ 被黑边和灰边连接，$1\leqslant x\leqslant n$。每一个这样由黑边和灰边组成的对定义了断点图中一个长度为 2 的圈。显而易见，存在一个 $G(\pi)$ 的最大圈分解，包含全部这些长度为 2 的 n 个圈。把有符号排列 $\vec{\pi}$ 的断点图 $G(\vec{\pi})$ 定义为除去这 $2n$ 条边的断点图 $G(\pi)$。注意到 $G(\vec{\pi})$ 中每个顶点的度为 2 [见图 10.9 (c)]，因此有符号排列的断点图是不相连圈的集合。记这种圈的数目为 $c(\vec{\pi})$。注意到 n 阶单位有符号排列映射到 $2n$ 阶等同（无符号）排列，$\vec{\pi}$ 上的反序

作用可以被 π 上的反序所模拟，于是就有 $d(\vec{\pi}) \geq d(\pi)$。

下面，通过将有符号排列 $\vec{\pi}$ 的像 $\pi=\pi_1\pi_2\cdots\pi_{2n}$ 进行排序，即按反序 $\rho(2i+1, 2j)$ 对 π 进行排序，而该反序仅仅在 π 的偶数位置（π_{2k-1} 和 π_{2k} 之间，$1\leq k\leq n$）之后“切割”。π 上的反序 $\rho(2i+1, 2j)$ 的作用可以被 $\vec{\pi}$ 上的反序 $\rho(i+1, j)$ 模拟，这就意味着如果 π_{2i-1} 和 π_{2i} 之间的切割是被禁止的，则 $d(\vec{\pi})=d(\pi)$。接下来，所考虑的无符号排列都是一些有符号排列的像。出于方便考虑，将“有符号排列”术语扩充到无符号排列 $\pi=(\pi_1\pi_2\cdots\pi_{2n})$，这样 π_{2i-1} 和 π_{2i} 是连续数，$1\leq i\leq n$。π 上的反序 $\rho(i, j)$ 是合法的，如果 i 是奇数，j 是偶数。注意在一个有符号的排列中的任一反序对应于其像上的合法反序，反之亦然，下面提到的反序都是指合法反序。

给定任意一个反序 ρ，记 $\Delta c\equiv\Delta c(\pi, \rho)=c(\pi\rho)-c(\pi)$（随圈分解的大小而增加）。定理 10.1 隐含对每一个排列 π 和反序 ρ，有 $\Delta c\equiv\Delta c(\pi, \rho)\leq 1$。如果 $\Delta c=1$，则称反序为恰当的。

如果能够对每个排列找到一个恰当反序，那么就可以在 $n+1-c(\pi)$ 步中对排列 π 进行最佳排序。但是，对排列 $\pi=+3+2+1$，没有恰当反序，因此，它不能在 $n+1-c(\pi)=2$ 步中完成排序（如图 10.10 所示为这个排列的最佳排序）。这指出除圈数目之外还存在另外的“障碍”影响根据反序的排序。排列 $\pi=+3+2+1$ 包含一个障碍（hurdle），表明这样的一个“隐藏”障碍会影响根据反序的排序。有关障碍的概念会在下一节中定义。

我们说反序 $\rho(i, j)$ 作用于 $G(\pi)$ 上的黑边 (π_{i-1}, π_i) 和 (π_j, π_{j+1})。如果黑边 (π_{i-1}, π_i) 和 (π_j, π_{j+1}) 属于 C，则 $\rho(i, j)$ 是 $G(\pi)$ 的圈 C 上的反序（起作用的）。如果作用于关联着 g 的两条黑边上的反序是恰当的，则灰边 g 是定向的，否则就是无定向的。例如，在图 10.9（c）中的灰边（8，9）和（22，23）是定向的，而灰边（4，5）和（18，19）是无定向的。

引理 10.3　设 (π_i, π_j) 是一条连接黑边 (π_k, π_i) 和 (π_j, π_l) 的灰边。则 (π_i, π_j) 是定向的，当且仅当 $i-k=j-l$。

$G(\pi)$ 中的圈是定向的，如果该圈有一条定向的灰边而其他的是无定向的。图 10.9（c）中的圈 C 和 F 是定向的，而圈 A，B，D，E 是无定向的。显而易见，不存在作用于一个无定向圈上的恰当反序。

10.6　交叉图和障碍

$G(\pi)$ 中的灰边 (π_i, π_j) 和 (π_k, π_t) 是交叉的，如果区间 $[i, j]$ 和 $[k, t]$ 交叠，但它们相互不包含。例如，图 10.9（c）中的边（4，5）和（18，19）是交叉的，而边（4，5）和（22，23）还有（4，5）和（16，17）是不交叉的。圈 C_1、C_2 是交叉的，如果存在交叉灰边 $g_1\in C_1$ 和 $g_2\in C_2$。

设 $\mathcal{C}_\pi$ 是排列 π 的断点图中圈的集合。定义 π 的一个交叉图 H_π（$\mathcal{C}_\pi$，$\mathcal{I}_\pi$），它带有边的集合

$$\mathcal{I}_\pi=\{(C_1,C_2):C_1 \text{ 和 } C_2 \text{ 是 } G(\pi)\text{中的交叉圈}\}。$$

如图 10.9(d)所示一个由 3 个连通分支(connected component)构成的交叉图 H_π。H_π 的顶点集被分为定向顶点和无定向顶点（$\mathcal{C}_\pi$ 中的圈）。H_π 的一个连通分支是定向的，如果它至少含有一个定向顶点，否则就是无定向的。对一个连通分支 U，定义 U 最左点和最右点为

$$U_{\min}=\min_{\pi_i\in C\in U} i \text{ 和 } U_{\max}=\max_{\pi_i\in C\in U} i$$

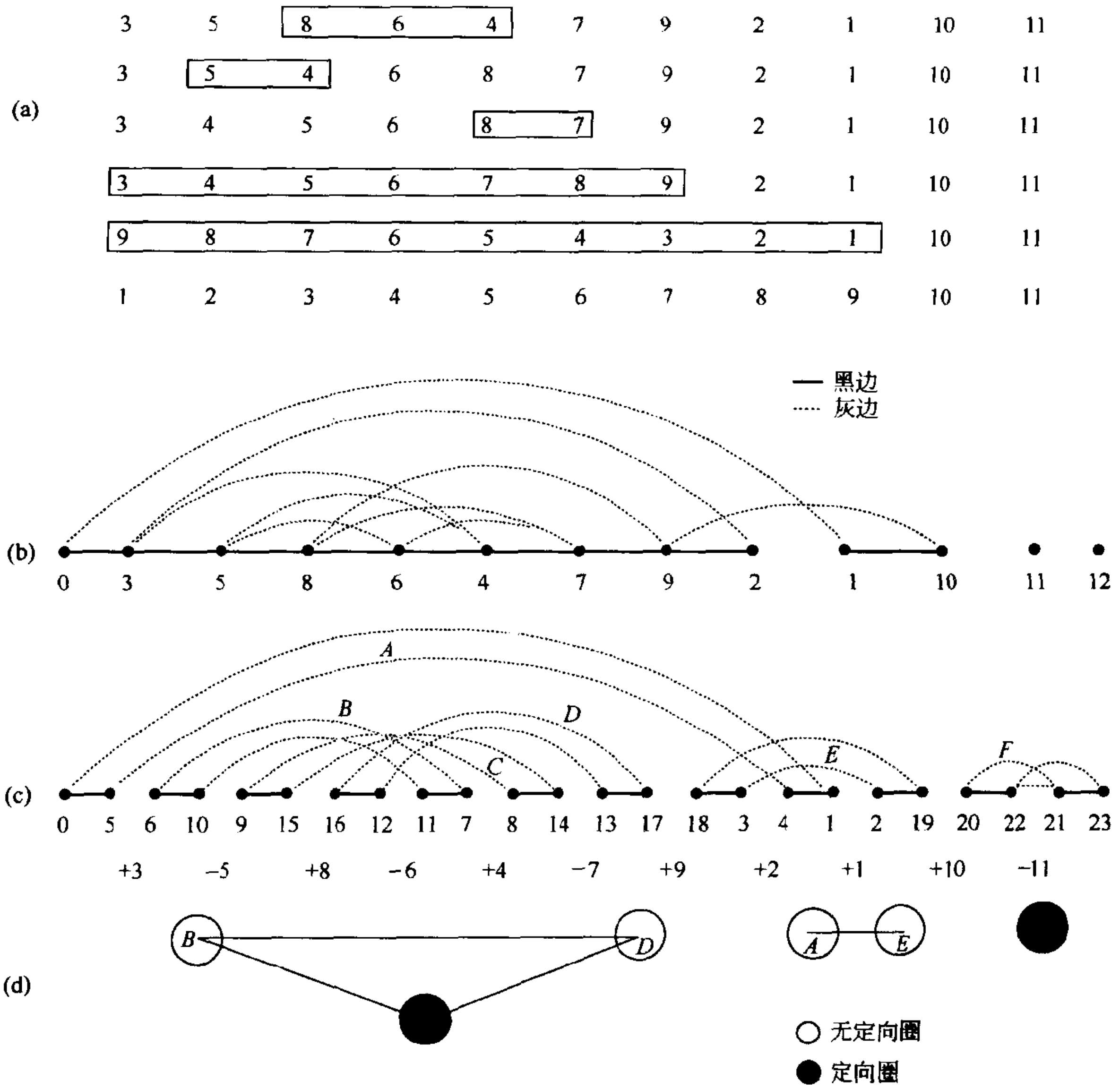


图 10.9 (a) 排列 (3 5 8 6 4 7 9 2 1 10 11) 经过 5 个反序的最佳排序；(b) 这个排列的断点图；(c) 一个有符号排列变换为一个无符号排列 π 和断点图 $G(\pi)$；(d) 带两个定向分支和一个无定向分支的交叉图 H_π

例如，图 10.9（c）中包含圈 B，C 和 D 的分支 U 的最左边的顶点 $\pi_2=6$，最右边的顶点 $\pi_{13}=17$；因此 $[U_{\min}, U_{\max}]=[2, 13]$。

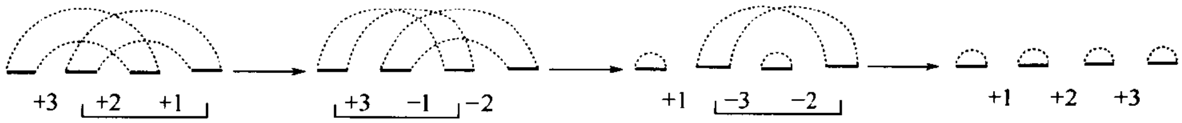


图 10.10 排列 $\pi=+1+2+3$ 的最佳排序包含了一个非恰当反序

我们说分支 U 在 π 中分离分支 U'和 U''，如果 U 中存在一条灰边 (π_i, π_j)，使得 $[U'_{\min}, U'_{\max}]\subset[i, j]$ 但 $[U''_{\min}, U''_{\max}]\not\subset[i, j]$。例如，图 10.11（a）中的分支 U 分离分支 U'和 U''。

记≺是集合 P 上的偏序。元素 $x\in P$ 称为在≺中的最小元素，如果不存在元素 $y\in P$ 使得 $y\prec x$。元素 $x\in P$ 称为在≺中的最大元素，如果对每个 $y\in P$ 有$y\prec x$。

考虑 H_π 中的无定向分支集 $\mathcal{U}_\pi$，在此集合上定义容积偏序（containment partial order），即，如果 $[U_{\min}, U_{\max}]\subset[W_{\min}, W_{\max}]$，$U$，$W\in\mathcal{U}_\pi$，则 $U\prec W$。一个障碍定义为一个无定向分支，不是最小障碍就是最大障碍，最小障碍 $U\in\mathcal{U}_\pi$是≺中的最小元素，最大障碍满足下面两个条件：（ⅰ）U 是≺中的最大元素；且（ⅱ）U 不分离任何两个障碍。设 $h(\pi)$ 是 π 中所有障碍的个数。图 10.9（c）中的排列 π 有一个无定向分支并且 $h(\pi)=1$。图 10.11（b）中的排列 π 有两个最小障碍和一个最大障碍 $[h(\pi)=3]$。图 10.11（a）中的排列 π 有两个最小障碍而没有最大障碍 $[h(\pi)=2]$，因为图 10.11（a）中的最大无定向分支 U 分离 U'和 U''。

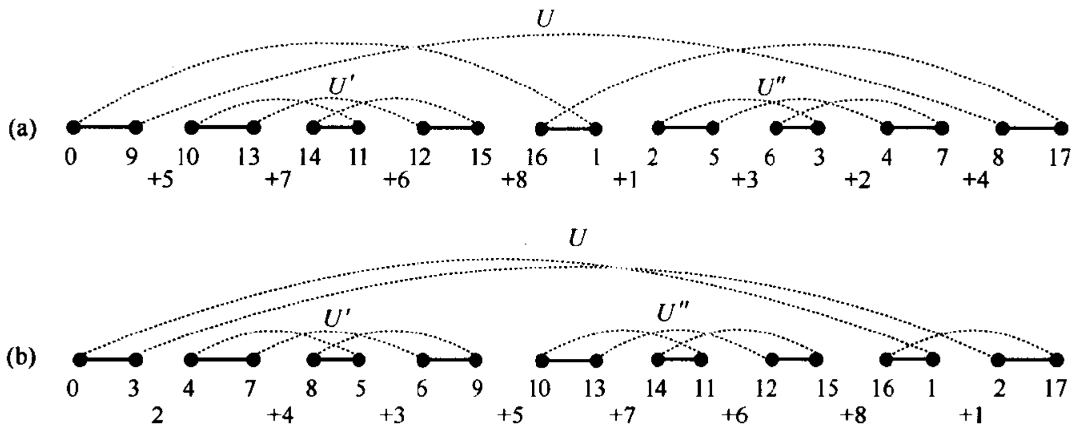


图 10.11 （a）无定向分支 U 通过边（0，1）分离 U'和 U''；
（b）障碍 U 不分离 U'和 U''

下面的定理进一步改进了根据反序对有符号排列进行排序的界限。

定理 10.6 对任意（有符号）排列 π，$d(\pi)\geqslant n+1-c(\pi)+h(\pi)$。

证明 给定一个任意反序 ρ，记 $\Delta h \equiv \Delta h(\pi, \rho) = h(\pi\rho) - h(\pi)$。显然，每个反序 ρ 作用于至多有两个障碍的黑边，因此 ρ 可能"破坏"至多两个最小障碍。注意到如果 ρ 在 $\mathcal{U}_\pi$ 中破坏了两个最小障碍，那么 ρ 不能破坏 $\mathcal{U}_\pi$ 中的最大障碍［参看最大障碍定义的条件（ⅱ）］。因此对每个 ρ，有 $\Delta h \geqslant -2$。 ■

定理 10.1 意味着 $\Delta c \in \{-1, 0, 1\}$。如果 $\Delta c = 1$，则 ρ 作用于一个定向圈，因此不影响 π 中任何障碍。所以 $\Delta h = 0$，且 $\Delta(c-h) \equiv \Delta c - \Delta h = 1$。如果 $\Delta c = 0$，那么 ρ 作用于圈，因此最多影响一个障碍［参看最大障碍定义的条件（ⅱ）］。这表示 $\Delta h \geqslant -1$，$\Delta(c-h) \leqslant 1$。如果 $\Delta c = -1$，那么 $\Delta(c-h) \leqslant 1$，因为对每个反序 ρ，$\Delta h \geqslant -2$。

因此，对一个任意反序 ρ，$\Delta(c-h) \leqslant 1$。又因为对一个恒等排列 l，$c(l) = n+1$，$h(l) = 0$，可知 $d(\pi) \geqslant [c(l) - h(l)] - [c(\pi) - h(\pi)] = n + 1 - c(\pi) + h(\pi)$ ■

Hannenhalli 和 Pevzner [154] 证明下界 $d(\pi) \geqslant n+1-c(\pi)+h(\pi)$ 是紧密的。作为得到上界 $d(\pi) \leqslant n+1-c(\pi)+h(\pi)+1$ 的第一步，发展了一种称为排列的等价变换（equivalent transformation）的技术。

10.7 排列的等价变换

断点图中长圈的复杂交叉结构给分析按反序排序带来了一系列困难。为避开这个问题，基于下面的思想，引进了排列的等价变换。如果排列 $\pi \equiv \pi(0)$ 有一个长圈，通过将这个长圈"断裂"成两个小一点的圈，可将排列 $\pi \equiv \pi(0)$ 变换成一个新的排列 $\pi(1)$。用同样的方法继续处理 $\pi(1)$，得到一个排列序列 $\pi \equiv \pi(0), \pi(1), \cdots, \pi(k) \equiv \sigma$，以一个简单（即没有长圈）排列结尾。本节证明了这些变换可以被排序，其思想是通过用每个 σ 的排序来模拟对 π 的具有相同反序个数的排序。下一节说明如何对简单排列进行最佳排序。利用简单排列 σ 的最佳排序来模拟任意排列 π 的最佳排序，可以得到按反序排序的多项式算法。

$C = \cdots, v_b, w_b, \cdots, w_g, v_g, \cdots$ 是排列 π 的断点图 $G(\pi)$ 中的圈，设 $b = (v_b, w_b)$ 和 $g = (w_g, v_g)$ 分别为其中的黑边和灰边。$G(\pi)$ 的 (g, b)-分裂(split）是一张新图，记为 $\hat{G}(\pi)$，可以通过以下方法从 $G(\pi)$ 中得到。

- 删除边 g，b。
- 增加两个新的顶点 v，w。
- 增加两条新的黑边 (v_b, v)，(w, w_b)。
- 增加两条新的灰边 (w_g, w)，(v, v_g)。

如图 10.12 所示一个把 $G(\pi)$ 中的圈 C 变换成 $\hat{G}(\pi)$ 中的圈 C_1，C_2 的 (g, b)-分裂。如果 $G(\pi)$ 是有符号排列 π 的断点图，那么 $G(\pi)$ 的每个 (g, b)-分裂

对应于一个有符号的广义排列(generalized permutation) $\hat{\pi}$ 的断点图，其中 $\hat{G}(\pi)=G(\hat{\pi})$。下面定义广义排列，并且描述如何使用填充程序来找到一个对应于 G 的 (g, b)-分裂的广义排列 $\hat{\pi}$。

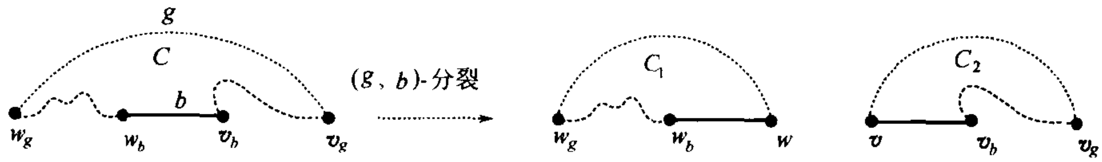


图 10.12 (g, b)-分裂的例子

广义排列 $\pi=\pi_1\pi_2\cdots\pi_n$ 是指任意各不相同实数组成的排列（代替整数 $\{1, 2, \cdots, n\}$ 组成的排列）。在这一节中，排列就是指广义排列，单位广义排列就是指广义排列 $\pi=\pi_1\pi_2\cdots\pi_n$ 满足 $\pi_i<\pi_{i+1}$，$1\leqslant i\leqslant n-1$。通过增加 $\pi_0=\min\limits_{1\leqslant i\leqslant n}\pi_i-1$，$\pi_{n+1}=\max\limits_{1\leqslant i\leqslant n}\pi_i+1$，扩充排列 $\pi=\pi_1\pi_2\cdots\pi_n$。如果不存在元素 π_l，使得 $\pi_j<\pi_l<\pi_k, 1\leqslant l\leqslant n$，则称 π 的元素 π_j，π_k 是连续的。π 的元素 π_i，π_{i+1} 是相邻的，$0\leqslant i\leqslant n$。(广义) 排列 $\pi=\pi_1\pi_2\cdots\pi_n$ 的断点图定义为连接顶点 $\{\pi_0, \pi_1, \cdots, \pi_n, \pi_{n+1}\}$ 上相邻元素（不连续）的黑边和连接连续元素（不相邻）的灰边所构成的图。显然，对于广义排列，断点图的定义与先前讨论的断点图概念是一致的。

$C=\cdots, \pi_{i+1}, \pi_i, \cdots, \pi_j, \pi_k, \cdots$ 是断点图 $G(\pi)$ 中的一个圈，设 $b=(\pi_{i+1}, \pi_i)$ 和 $g=(\pi_j, \pi_k)$ 分别是其中的黑边和灰边。定义 $\Delta=\pi_k-\pi_j$ 并设 $v=\pi_j+\frac{\Delta}{3}$，$w=\pi_k-\frac{\Delta}{3}$。一个 $\pi=(\pi_1\pi_2\cdots\pi_n)$ 的 (g, b)-填充(padding) 是 $n+2$ 个元素的排列，通过在 π 的第 i 个元素 $(0\leqslant i\leqslant n)$ 之后插入 v，w 得到

$$\hat{\pi}=\pi_1\pi_2\cdots\pi_i v w \pi_{i+1}\cdots\pi_n$$

注意到 v 和 w 在 $\hat{\pi}$ 中不仅连续而且相邻，这就意味着如果 π 是符号排列(的像)，那么 $\hat{\pi}$ 也是符号排列（的像)。下面的引理确定 (g, b)-填充和(g, b)-分裂之间的对应关系❶。

引理 10.4 $\hat{G}(\pi)=G(\hat{\pi})$。

如果 g 和 b 是 $G(\pi)$ 中一个长圈 C 的不相连边，那么 (g, b)-填充把 C 分割为$G(\hat{\pi})$ 中的两个较小圈。因此填充可以将任意排列 π 变换成一个简单排列。注意到 $\hat{\pi}$ 中有 $\hat{n}=n+1$ 个元素并且$c(\hat{\pi})=c(\pi)+1$。下面证明对每一个有长圈的排列，其圈的不相连边上存在一个填充，使得 $h(\hat{\pi})=h(\pi)$，这就意味着填充提供了一种不改变参数 $n+1-c(\pi)+h(\pi)$ 而消除排列中长圈的方法。首先，

---

❶ 当然，如果 $\pi_i>\min\{\pi_j, \pi_k\}$，则在 $\{1, 2, \cdots, n\}$上的排列 $\pi=(\pi_1\pi_2\cdots\pi_n)$ 的 (g, b)-填充可以被模拟为在 $\{1, 2, \cdots, n+2\}$ 上的排列 $\hat{\pi}=(\hat{\pi}_1\hat{\pi}_2\cdots\hat{\pi}_i vw\hat{\pi}_{i+1}\cdots\hat{\pi}_n)$，其中 $v=\pi_j+1$，$w=\pi_k+1$，$\hat{\pi}_i=\pi_i+2$，否则$\hat{\pi}_i=\pi_i$。引入广义排列可使下面的“模拟”过程更直观。

需要一系列技巧方面的引理。

引理 10.5 设 $G(\pi)$ 中某一圈 C 上的 (g, b)-填充删除灰边 g，并增加两个新的灰边 g_1 和 g_2。如果 g 是定向的，那么 g_1 或 g_2 在 $G(\hat{\pi})$ 中是定向的。如果 C 是无定向的，g_1 和 g_2 在 $G(\hat{\pi})$ 中都无定向。

引理 10.6 设 (g, b)-填充把 $G(\pi)$ 中的圈 C 断裂成 $G(\hat{\pi})$ 中的圈 C_1 和 C_2。C 是定向的，当且仅当 C_1 或 C_2 是定向的。

证明 注意到 (g, b)-填充保持 $G(\hat{\pi})$ 中灰边从 $G(\pi)$ “继承”得到的方向（引理 10.3）。如果 C 是定向的，那么有一条定向灰边。如果这条边不同于 g，那么它在 π 的 (g, b)-填充中保持定向，因此一个包含这条边的圈（C_1 或 C_2）是定向的。如果 $g=(w_g, v_g)$ 是 C 中惟一的定向灰边，那么 (g, b)-填充在 $G(\hat{\pi})$ 中增加两条新的灰边 $[(w_g, w), (v, v_g)]$，其中一条是定向的（引理 10.5）。因此包含这条边的圈（C_1 或 C_2）是定向的。

如果 C 是无定向圈，那么所有从 C 继承的 C_1，C_2 的边保持无定向。由引理 10.5 可知，C_1 和 C_2 中的新边 $[(w_g, w)$ 和 $(v, v_g)]$ 也无定向。 ■

下面的引理显示填充保持灰边的交叉。

引理 10.7 设 g' 和 g'' 是 $G(\pi)$ 的两条不同于 g 的灰边。那么 g' 和 g'' 在 π 中是交叉的，当且仅当 g' 和 g'' 在 π 的 (g, b)-填充中是交叉的。

由这个引理立即推出以下引理。

引理 10.8 设 (g, b)-填充将 $G(\pi)$ 中的圈 C 断裂成 $G(\hat{\pi})$ 中的圈 C_1 和 C_2。每个在 $G(\pi)$ 中与 C 交叉的圈 D 与 $G(\pi)$ 中的 C_1 或 C_2 交叉。

证明 设 $d \in D$，$c \in C$ 是 $G(\pi)$ 中的交叉灰边。如果 c 不同于 g，那么由引理 10.7 可得 d 和 c 在 $G(\hat{\pi})$ 中交叉，因此 D 与 C_1 或 C_2 交叉。如果 $c=g$，那么很容易在 $G(\hat{\pi})$ 中找到一条与 d 交叉的新的灰边，因此 D 与 C_1 或 C_2 交叉。 ■

引理 10.9 对每条灰边 g，在 $G(\pi)$ 中存在与 g 交叉的灰边 f。

引理 10.10 设 C 是 $G(\pi)$ 中的圈，$g \notin C$ 是 $G(\pi)$ 中的一条灰边。那么 g 与 C 中的偶数条灰边交叉。

一个 (g, b)-填充 ϕ 把 π 变换成 $\hat{\pi}$（即 $\hat{\pi}=\pi \cdot \phi$）是安全的，如果它作用于一个长圈的不相连边上，并且 $h(\pi)=h(\hat{\pi})$。显而易见，每个安全的填充把一个长圈断裂成两个较小的圈。

定理 10.7 如果 C 是 $G(\pi)$ 中的一个长圈，那么存在一个作用于 C 的安全 (g, b)-填充。

证明 如果 C 有一对交叉灰边 $g_1, g_2 \in C$，那么去掉这些边会将 C 变换成两条路。因为 C 是长圈，所以至少一条路包含灰边 g。从另一路中挑选黑边 b，考虑将 π 变换成 $\hat{\pi}$ 的 (g,b)-填充（显然，g 和 b 是不相连的边）。这个 (g,b)-填充

把 C 断裂成 $G(\hat{\pi})$ 中的 C_1 和 C_2，g_1 和 g_2 分别属于不同的圈 C_1 和圈 C_2。根据引理 10.7，g_1 和 g_2 是交叉的，这就意味着 C_1 和 C_2 是交叉的。同样，根据引理 10.8，这个 (g,b)-填充并不"断裂" H_π 中包含圈 C 的分支 K，所有来自 K 的圈都属于包含 C_1 和 C_2 的 H_π 中的分支。而且，根据引理 10.6，这个分支在 H_π 中的方向和在 $H_{\hat{\pi}}$ 中的一致。因此这个被选中的 (g,b)-填充保持了障碍集，且 $h(\pi)=h(\hat{\pi})$。

如果 C 中全部的灰边彼此不交叉，则 C 是一个无定向圈。由引理 10.9 和引理 10.10 可知，存在一条灰边 $e\in C'$，与至少两条灰边 $g_1,g_2\in C$ 交叉。去掉 g_1 和 g_2 把 C 变换成两条路。由于 C 是一个长圈，这些路中至少有一条包含灰边 g。从另一条路中挑选一条黑边，并考虑 π 的 (g,b)-填充。这个填充把 C 分成 $G(\hat{\pi})$ 中的圈 C_1 和圈 C_2，g_1 和 g_2 分别属于不同的圈 C_1 和圈 C_2。根据引理 10.7，C_1 和 C_2 在 $\hat{\pi}$ 中与 C' 交叉。因此，这个 (g,b)-填充不断裂 H_π 中包含 C_1 和 C_2 的分支 K。此外，根据引理 10.6，C_1 和 C_2 是无定向的，这就意味着该分支在 H_π 和 $H_{\hat{\pi}}$ 中的方向一致。所以，选定的 (g,b)-填充保持了障碍集，因此 $h(\pi)=h(\hat{\pi})$。 ■

一个排列 π 等价于排列 $\sigma(\pi \rightsquigarrow \sigma)$，如果存在一系列的排列 $\pi\equiv\pi(0)$，$\pi(1),\cdots,\pi(k)\equiv\sigma$，使得对作用于 $\pi_i(0\leqslant i\leqslant k-1)$ 的一个安全的 (g,b)-填充满足 $\pi(i+1)=\pi(i)\phi(i)$。

定理 10.8 对每个排列，存在一个等价的简单排列。

证明 排列 π 的复杂度定义为 $\sum_{C\in\mathcal{C}_\pi}[l(C)-2]$，其中 $\mathcal{C}_\pi$ 是 $G(\pi)$ 中的圈集，$l(C)$ 是圈 C 的长度。简单排列的复杂度为 0。注意到长圈 C 的不相连边上的每个填充把圈 C 断裂成圈 C_1 和圈 C_2，且 $l(C)=l(C_1)+l(C_2)-1$。因此

$$[l(C)-2]=[l(C_1)-2]+[l(C_2)-2]+1$$

这表明圈的不相连边上的填充减少了排列的复杂度。这个结果和定理 10.7 说明，通过一系列的保持 $b(\pi)-c(\pi)+h(\pi)$ 的填充，每个有长圈的排列可以被变换成没有长圈的排列。 ■

设 $\hat{\pi}$ 是 π 的一个 (g,b)-填充，ρ 是作用于 $\hat{\pi}$ 的两条黑边上的反序，那么通过忽略填充的元素可以在 π 上模拟 ρ，这种观察的一般性描述是需要的。一个广义排列的序列 $\pi\equiv\pi(0),\pi(1),\cdots,\pi(k)\equiv\sigma$ 被称为 π 的广义排序，如果 σ 是单位（广义）排列，且 $\pi(i+1)$ 通过反序或填充从 $\pi(i)$ 得到。注意，在 π 的广义排序中的反序和填充可能交叉。

引理 10.11 每个 π 的广义排序模拟了 π 的一个（真实的）排序，两者的反序数相同。

证明 忽略填充元素。 ■

下面，说明如何通过一系列填充和包含 $d(\pi)$ 个反序的反序，来发现排列 π

的一个广义排序。引理 10.11 表明了这种 π 的广义排序模拟了 π 的最佳（真实的）排序。

10.8 搜索安全反序

回想对一个任意反序，$\Delta(c-h)\leqslant 1$（参看定理 10.6 的证明）。如果 $\Delta(c-h)=1$，则反序 ρ 是安全的。图 10.10 中的第一个反序不是恰当的但是安全的（因为 $\Delta c=0$，$\Delta h=-1$）。图 10.14 显示了安全反序（$\Delta c=1$，$\Delta h=0$）和不安全反序（$\Delta c=1$，$\Delta h=1$）的例子。下面，通过分析反序在简单排列上的作用，证明定向分支中存在一个作用于圈上的安全反序。本节中的圈均指短圈，排列均指简单排列。

与 $G(\pi)$ 中圈 C 交叉的所有圈的集合记为 $V(C)$ [即 $V(C)$ 是 H_π 中与 C 相邻顶点的集合]。由 $V(C)$ 导出的 H_π 的子图中的边的集合记为

$$E(C)=\{(C_1,C_2):C_1,C_2\in V(C) \quad \text{在 } \pi \text{ 中 } C_1 \text{ 与 } C_2 \text{ 交叉}\}$$

它的补集记为

$$\overline{E}(C)=\{(C_1,C_2):C_1,C_2\in V(C) \quad \text{在 } \pi \text{ 中 } C_1 \text{ 与 } C_2 \text{ 不交叉}\}$$

作用于一个定向（短）圈 C 上的反序 ρ “破坏” C [即从 $G(\pi)$ 中删除 C 的边] 并且把 $G(\pi)$ 中的其他的圈变换成 $G(\pi\rho)$ 中相同顶点上的相应的圈。结果，ρ 把 π 的交叉图 $H_\pi(\mathcal{C}_\pi,\mathcal{I}_\pi)$ 变换成 $\pi\rho$ 的交叉图 $H_{\pi\rho}(\mathcal{C}_\pi\backslash C,\mathcal{I}_{\pi\rho})$。如下面的引理（见图 10.13）所述，这个变换得到由 $V(C)$ 导出的子图的补集。我们记 $\overline{\mathcal{I}}_{\pi\rho}=\mathcal{I}_\pi\backslash\{(C,D):D\in V(C)\}$。

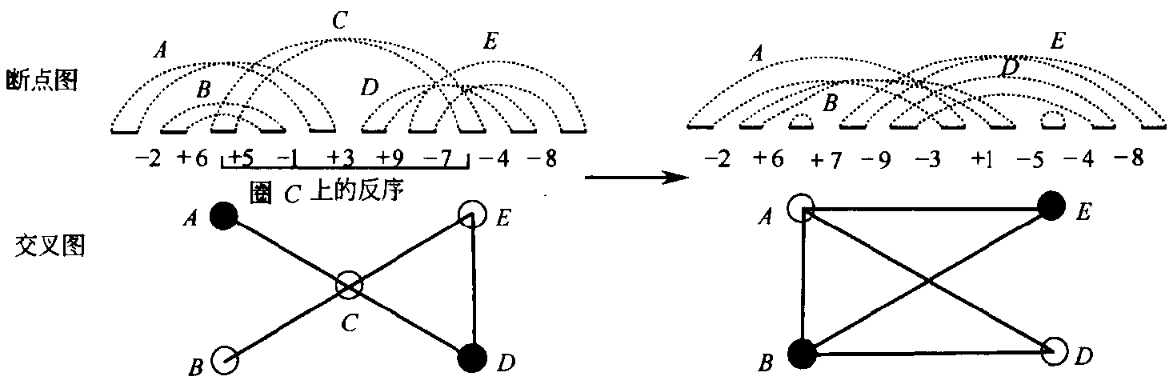


图 10.13 圈 C 上的反序补充了 C 相邻之间的边，并改变了交叉图中每个和 C 相邻的圈的定向

引理 10.12 设 ρ 是作用于一个定向（短）圈 C 上的反序。那么

- $\mathcal{I}_{\pi\rho}=(\overline{\mathcal{I}}_\pi\backslash E(C))\cup\overline{E}(C)$。即 ρ 删除边 $E(C)$ 并添加边 $\overline{E}(C)$，将 H_π 变换成 $H_{\pi\rho}$。
- ρ 改变圈 $D\in\mathcal{C}_\pi$ 的定向，当且仅当 $D\in V(C)$。

从引理 10.12 立即可以推出下面的引理。

引理 10.13 设 ρ 是作用于圈 C 上的反序，A 和 B 是 $H_{\pi\rho}$ 中非相邻的顶点。那么 (A,B) 是 H_π 中的一条边，当且仅当 $A,B\in V(C)$

设 K 是 H_π 的一个定向分支，$\mathcal{R}(K)$ 是作用于 K 中的定向圈上的反序的集合。假定反序 $\rho\in\mathcal{R}(K)$ 把 K“断裂”成 $H_{\pi\rho}$ 中许多相连的分支 $K_1(\rho)$，$K_2(\rho),\cdots$，且这些分支的前 m 个是无定向的。如果 $m>0$，那么 ρ 可能是不安全的，因为分支 $K_1(\rho),\cdots,K_m(\rho)$ 可能在 $\pi\rho$ 中形成新的障碍，这样和 $h(\pi)$ 相比 $h(\pi\rho)$ 增大（见图 10.14）。下面，将说明从集合 $\mathcal{R}(K)$ 中选择反序存在一定的灵活性，允许用一个安全反序 σ 替换不安全反序 ρ。

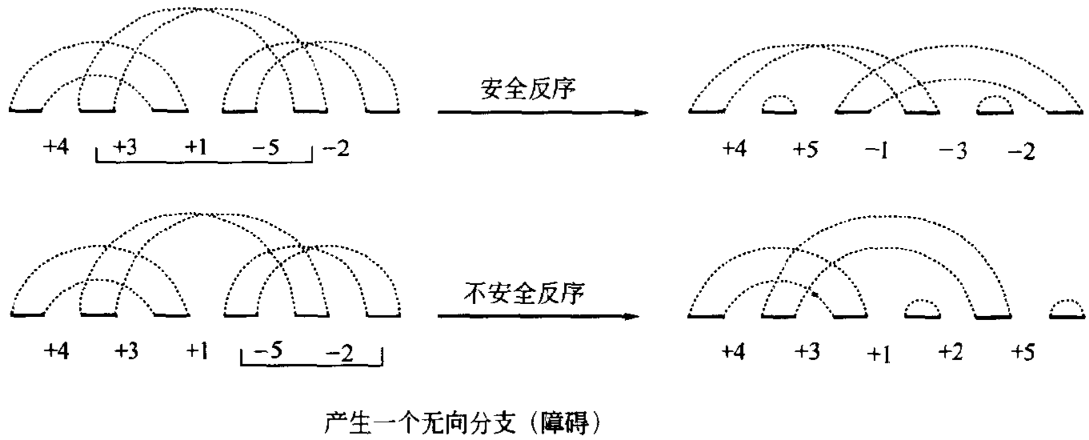


图 10.14 安全反序和不安全反序的例子

引理 10.14 设 ρ 和 σ 是分别作用于 $G(\pi)$ 中两个交叉定向圈 C 和 C' 上的反序。如果 C' 属于 $H_{\pi\rho}$ 中的一个无定向分支 $K_1(\rho)$，那么

- $K_1(\rho)$ 外面的每两个顶点如果在 $H_{\pi\rho}$ 中相邻，则它们在 $H_{\pi\sigma}$ 中也相邻。
- 与 $H_{\pi\rho}$ 相比，在 $H_{\pi\sigma}$ 中 $K_1(\rho)$ 外部的顶点的定向并不改变。

证明 设在 $H_{\pi\rho}$ 中，D 和 E 是 $K_1(\rho)$ 外部的由一条边相连的两个顶点。如果其中一个顶点，比方说 D，不属于 H_π 中的 $V(C)$，那么由引理 10.13 可知（ⅰ）(C',D) 不是 H_π 中的边；（ⅱ）(D,E) 是 H_π 中的边。因此，根据引理 10.12，反序 σ 保持 $H_{\pi\sigma}$ 中的边 (D,E)。如果顶点 D，E 都属于 $V(C)$，那么根据引理 10.12 推出 (D,E) 不是 H_π 中的边。因为顶点 C' 和顶点 D,E 在 $H_{\pi\rho}$ 的不同分支中，那么由引理 10.13 推出 (C',D) 和 (C',E) 是 H_π 中的边。因此，根据引理 10.12，(D,E) 是 $H_{\pi\sigma}$ 中的一条边。在两种情况下，σ 保持了在 $H_{\pi\sigma}$ 中的边 (D,E)。引理的第一部分成立。

由引理 10.13 可知，对每个 $K_1(\rho)$ 外部的顶点 D，有 $D\in V(C)$，当且仅当 $D\in V(C')$。由此观察以及引理 10.12，可推得与 $H_{\pi\rho}$ 相比，在 $H_{\pi\sigma}$ 中 $K_1(\rho)$ 外部的顶点的定向不改变。 ■

引理 10.15 （简单排列的）交叉图中，每一个无定向分支包含至少两个顶点。

证明 根据引理 10.9，$G(\pi)$中的每条灰边都有一条灰边与之交叉。因此，$G(\pi)$中每一个无定向（短）圈有一个交叉圈。 ■

定理 10.9 对 H_π 中的每一个定向分支 K，都存在一个（安全的）反序 $\rho\in\mathscr{R}(K)$，使得所有的分支 $K_1(\rho),K_2(\rho),\cdots$在 $H_{\pi\rho}$中定向。

证明 假定反序 $\rho\in\mathscr{R}(K)$把 K“断裂”成许多在 $H_{\pi\rho}$中相连的分支 $K_1(\rho)$，$K_2(\rho),\cdots$,且这些分支的前 m 个是无定向的。记这些无定向分支中的顶点总数为 $index(\rho)=\sum_{i-1}^{m}|K_i(\rho)|$，其中$|K_i(p)|$是 $K_i(\rho)$中的顶点数。设 ρ 是反序，使得

$$index(\rho)=\min_{\sigma\in\mathscr{R}(K)} index(\sigma)$$

这个反序作用于圈 C 上并且把 K 断裂成许多分支。如果这些分支都是定向的[即 $index(\rho)=0$]，定理成立。否则，$index(\rho)>0$，并设 $K_1(\rho),\cdots,K_m(\rho)(m\geqslant1)$是 $H_{\pi\rho}$中的无定向分支。下面，找出另一个反序 $\sigma\in\mathscr{R}(K)$，使得 $index(\sigma)<index(\rho)$，矛盾。

设 V_1 是 $H_{\pi\rho}$中的分支 $K_1(\rho)$的顶点集。注意到 $K_1(\rho)$至少包含一个来自 $V(C)$的顶点，考虑 H_π 中与 C 相邻的分支 $K_1(\rho)$的（非空）顶点集 $V=V_1\cap V(C)$。因为 $K_1(\rho)$是 $\pi\rho$ 中的无定向分支，所有来自 V 的圈在 π 中定向，所有来自 $V_1\setminus V$ 的圈在 π 中无定向（引理 10.12）。设 C'是 V 中的（定向）圈，σ 是作用于$G(\pi)$中 C'上的反序。由引理 10.14 可得，对 $i\geqslant2$，$H_{\pi\rho}$中的分支 $K_i(\rho)$的所有边在 $H_{\pi\sigma}$中被保留，同时与 $H_{\pi\rho}$相比，在 $H_{\pi\sigma}$中 $K_i(\rho)$中顶点的方向不改变。因此，$\pi\rho$ 的全部无定向分支 $K_{m+1}(\rho),K_{m+2}(\rho)$，…在 $\pi\sigma$ 中“幸存”，并且

$$index(\sigma)\leqslant index(\rho)$$

下面证明存在一个作用于来自 V 的圈上的反序 σ 使得 $index(\sigma)<index(\rho)$，矛盾。

如果 $V_1\neq V(C)$，在 $G(\pi)$中（定向）圈 $C'\in V$ 和（无定向）圈 $C''\in V_1\setminus V$ 之间存在一条边。由引理 10.12 可知，作用于 π 中 C'上的反序 σ 确定了 $G(\pi)$中圈 C''的定向。这个结果和引理 10.14 推出 σ 使 $index(\sigma)$至少比 $index(\rho)$小 1，矛盾。

如果 $V_1=V(C)$（K_1 的所有圈和 C 交叉），那么在 $V(C)$中存在至少两个顶点（引理 10.15）。更进一步，存在（定向）圈 $C',C''\in V_1$，使得(C',C'')在 π 中不交叉［否则，由引理 10.12 将得到 $K_1(\rho)$没有边，与 $K_1(\rho)$的连通性矛盾］。把 σ 定义为作用于 C'上的反序。由引理 10.12 可知 σ 保持 C''的方向，于是与 $index(\rho)$比较 $index(\sigma)$至少小 1，矛盾。

上面的讨论表明存在一个反序 $\rho\in\mathscr{R}(K)$，使得 $index(\rho)=0$，即 ρ 不产生新

的无定向分支。那么 $\Delta c(\pi,\rho)=1$ 且 $\Delta h(\pi,\rho)=0$，推出 ρ 是安全的。 ■

10.9 清除障碍

如果 π 包含一个定向分支，那么由定理 10.9 可知 π 中存在一个安全反序。在这一节中，讨论在缺乏任何定向分支的情况下寻找安全反序。设 $\prec$ 是集 P 上的偏序。称 P 中的 y 覆盖 x，如果 $x\prec y$ 且不存在元素 $z\in P$，使得 $x\prec z\prec y$。$\prec$ 的覆盖图 Ω 是（无向）图，具有顶点集 P 和边集 $\{(x,y):x,y\in P,y$ 覆盖 $x\}$。

设 $\mathcal{U}_\pi$ 是 H_π 中无定向分支的集合，并设 $[U_{\min},U_{\max}]$ 是无定向分支 $U\in\mathcal{U}_\pi$ 中最左位置和最右位置之间的间隔区间。定义 $\bar{U}_{\min}\doteq\min_{U\in\mathcal{U}_\pi}U_{\min}$，$\bar{U}_{\max}=\max_{U\in\mathcal{U}_\pi}U_{\max}$，并设 $[\bar{U}_{\min},\bar{U}_{\max}]$ 是在全部 π 的无定向分支中的最左位置和最右位置之间的间隔区间。设 $\bar{U}$ 是与区间 $[\bar{U}_{\min},\bar{U}_{\max}]$ 相关联的（假）分支。

定义 $\bar{\mathcal{U}}_\pi$ 是由 $|\mathcal{U}_\pi|$ 个元素 $\{U:U\in\mathcal{U}_\pi\}$ 和一个增加的元素 $\bar{U}$ 组成的，有 $|\mathcal{U}_\pi|+1$ 个元素的集合。设 $\prec\equiv\prec_\pi$ 是 $\bar{\mathcal{U}}_\pi$ 上的容积偏序，定义为规则 $U\prec W$ 当且仅当对 $U,W\in\bar{\mathcal{U}}_\pi$ 有 $[U_{\min},U_{\max}]\subset[W_{\min},W_{\max}]$。如果 π 中存在一个最大的无定向分支 U（即 $[U_{\min},U_{\max}]\subset[\bar{U}_{\min},\bar{U}_{\max}]$），假定存在两个元素（"真"分支 U 和"假"分支 $\bar{U}$）对应最大区间并且 $U\prec_\pi\bar{U}$。设 Ω_π 是树，表示 $\bar{\mathcal{U}}_\pi$ 上偏序 $\prec_\pi$ 的覆盖图［见图 10.15（a)］。Ω_π 中的每一个顶点除去 $\bar{U}$ 都与 $\mathcal{U}_\pi$ 中的一个无定向分支相联系。在 π 有最大障碍的情况下，假定叶子 $\bar{U}$ 与这个最大障碍相联系（即在这种情况下，有两个顶点和最大的障碍相对应，叶子 $\bar{U}$ 和它的相邻，最大的障碍 $U\in\mathcal{U}_\pi$)。Ω_π 上每一片与 $\prec_\pi$ 中最小元素相对应的叶子是一个障碍。如果 $\bar{U}$ 是 Ω_π 上的叶子，那它不一定是一个障碍（例如，$\bar{U}$ 是 Ω_π 上的叶子，但不是如图 10.11（a）所示的排列 π 的一个障碍)。因此，Ω_π 上叶子的数目和 $h(\pi)$ 中障碍的数目一致，除非

• 在 π 仅存在一个无定向分支［在这种情况下，Ω_π 由这个分支的两个拷贝构成，且有两片叶子，当 $h(\pi)=1$］。

• 在 $\mathcal{U}_\pi$ 中存在不是障碍的最大元素：即这个元素分割其他障碍［在这种情况下，叶子的数目等于 $h(\pi)+1$］。

每个障碍可以通过自身的任一圈上的反序变换成一个定向分支（见图 10.10)。这样一个操作"切断"覆盖图中的一片叶子，如下列引理所述。

引理 10.16（障碍切割，hurdle cutting）在障碍 K 中圈上的每个反序 ρ 从 π 的覆盖图中切断叶子 K，即 $\Omega_{\pi\rho}=\Omega_\pi\backslash K$。

证明 如果ρ作用于π中分支K的一个无定向圈，那么K在$\pi\rho$中保持“完整”。同样，由引理10.9可知，在（无定向）分支K中的（无定向）圈上的每个反序在K中至少定向一个圈。因此，ρ把K变换成$\pi\rho$中的一个定向分支并且从覆盖图中删除叶子K。 ■

切断障碍的反序不都是安全的。障碍$K \in \mathcal{U}_\pi$保护一个非障碍（non-hurdle）$U \in \mathcal{U}_\pi$，如果从$\mathcal{U}_\pi$中删除K将使U从非障碍变换成障碍（即U是$\mathcal{U}_\pi \setminus K$中的障碍）。$\pi$的一个障碍称为超障碍（superhurdle），如果它保护一个非障碍$U \in \mathcal{U}_\pi$，否则就称为简单障碍（simple hurdle）。图10.15（a）中的分支M, N和U是简单障碍，而分支L是超障碍（通过删除L把非障碍K变换成障碍）。在图10.16（a）中，三个障碍全部都是超障碍，而在图10.16（b）中有两个是超障碍，一个是简单障碍［注意在图10.16（a）和图10.16（b）中覆盖图是一样的！］。下面的引理可以由简单障碍的定义直接得出。

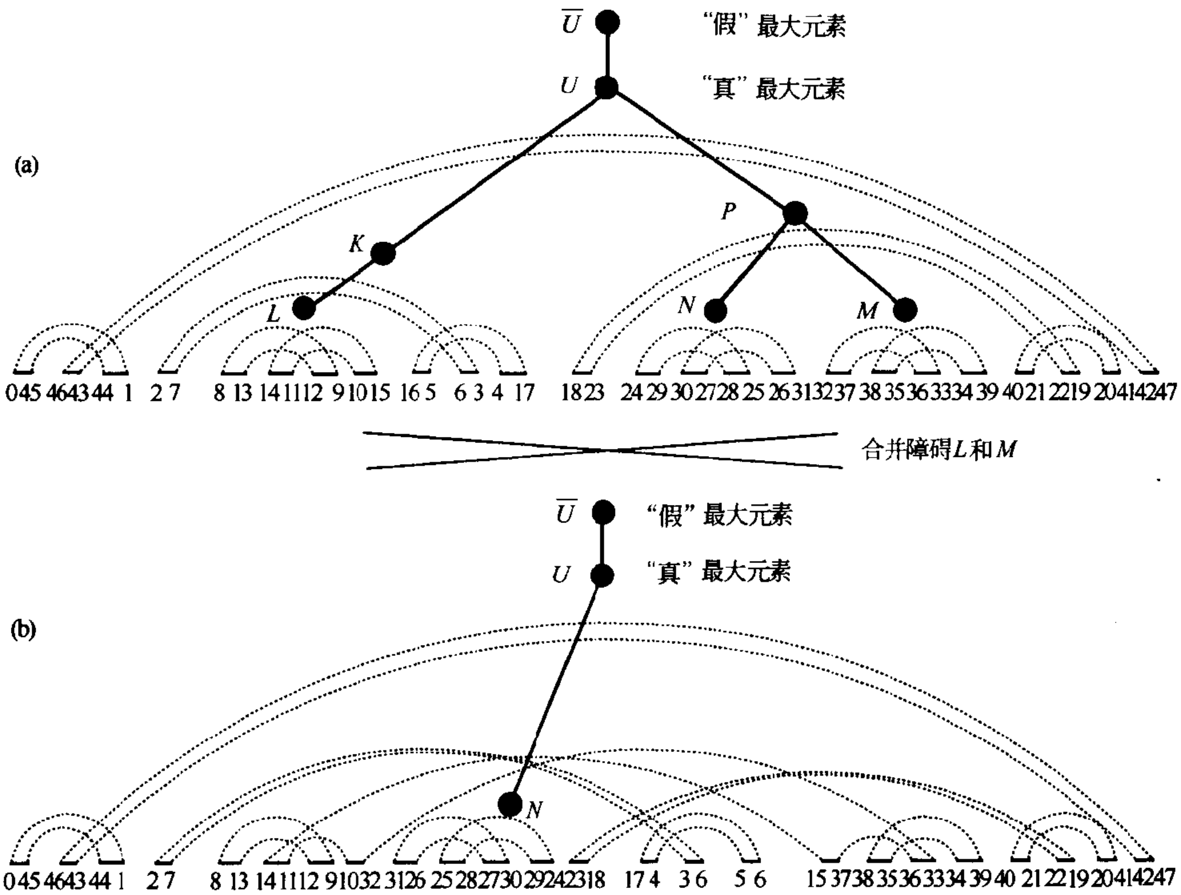


图10.15 （a）排列π的覆盖图Ω_π，具有“真”无定向分支K，L，M，N，P和U，及一个“假”分支$\bar{U}$；（b）一个反序ρ合并π中的障碍L和M，将无定向分支L，K，P和M变换为一个从$\Omega_{\pi\rho}$中“消失”的定向分支。这个反序将π中的无定向圈（32，33，36，37，32）和（10，11，14，15，10）变换为$\pi\rho$中的定向圈（15，14，11，10，32，33，36，37，15）。$LCA(L, M) = \overline{LCA}(L, M) = U$且$PATH(A, F) = \{L, K, U, P, M\}$。

引理 10.17 作用于简单障碍的圈的反序是安全的。

证明 由引理 10.16 可知对每个作用于简单障碍的圈的反序 ρ,$b(\pi)=b(\pi\rho)$,$c(\pi)=c(\pi\rho)$,$h(\pi\rho)=h(\pi)-1$,因此推出 ρ 是安全的。 ■

不幸的是，作用于超障碍的圈上的反序是不安全的，因为它把非障碍变换成障碍，可得 $\Delta(c-h)=0$。下面定义一个新的操作（障碍合并，hurdles merging），允许我们即使在缺乏简单障碍的情况下寻找安全反序。

如果 L 和 M 是 π 中的两个障碍，定义 $PATH(L,M)$ 为来自覆盖图 Ω_π 中叶子 L 到叶子 M 的（惟一）路径上的（无定向）分支集合。如果 L,M 都是 $\prec$ 中最小元素，定义 $LCA(L,M)$ 为一个 L 和 M 最小共同祖先的（无定向）分支，定义 $\overline{LCA}(L,M)$ 为不分离 L 和 M 的最小共同祖先。如果 L 对应最大障碍 U，那么在 $\bar{\mathcal{U}}_\pi$ 中存在对应于同样（最大）区间 $[U_{\min},U_{\max}]\subset[\bar{U}_{\min},\bar{U}_{\max}]$ 的两个元素 U 和 $\bar{U}$。在这种情况下，定义 $LCA(L,M)=\overline{LCA}(L,M)=U$。设 $G(V,E)$ 是图，$w\in V$,$W\subset V$。G 中 W 到 w 的收缩（contraction）定义为一张新图，其顶点集 $V\backslash(W\backslash w)$，边集 $\{[p(x),p(y)]:(x,y)\in E\}$，其中 $p(v)=w$，若 $v\in W$，否则 $p(v)=v$。注意，如果 $w\in W$，那么收缩减少 G 中顶点的个数为 $|W|-1$，如果 $w\notin W$，顶点减少的个数为 $|W|$。

设 L 和 M 是 π 中两个障碍，Ω_π 是 π 的覆盖图。定义 $\Omega_\pi(L,M)$ 为从 Ω_π 通过把 $PATH(L,M)$ 收缩成 $\overline{LCA}(L,M)$ 而得到的图［忽略图 $\Omega_\pi(L,M)$ 中的环］。注意，当

$$LCA(L,M)=\overline{LCA}(L,M)$$

时，$\Omega_\pi(L,M)$ 对应于从偏序 $\prec_\pi$ 中删除 $PATH(L,M)\backslash LCA(L,M)$ 的元素，而当

$$LCA(L,M)\neq\overline{LCA}(L,M)$$

时，$\Omega_\pi(L,M)$ 对应于从 $\prec_\pi$ 中删除整个 $PATH(L,M)$ 集合。

引理 10.18 （障碍合并）设 π 是带有覆盖图 Ω_π 的排列，ρ 是作用于 π 中（不同）障碍 L 和 M 的黑边上的反序。那么 ρ 作用于 Ω_π 上就像 $PATH(L,M)$ 收缩为 $\overline{LCA}(L,M)$ 一样，即 $\Omega_{\pi\rho}=\Omega_\pi(L,M)$。

证明 反序 ρ 作用于 $G(\pi)$ 中的圈 $C_1\in L$ 和 $C_2\in M$ 的黑边上，并把 C_1 和 C_2 变换成 $G(\pi\rho)$ 中的定向圈 C（见图 10.15）。容易证实每个和 $G(\pi)$ 中的 C_1 或 C_2 交叉的圈与 $G(\pi\rho)$ 中的 C 交叉。这表明 ρ 把 π 中的障碍 L 和 M 变换成 $\pi\rho$ 中一个定向分支的一部分，因此 L 和 M 从 $\Omega_{\pi\rho}$ 中“消失”。

而且，$PATH(L,M)\backslash\overline{LCA}(L,M)$ 的每个（无定向）分支至少有一个圈与 $G(\pi\rho)$ 中的 C 交叉。这表明在 π 中每个这种分支成为 $\pi\rho$ 中一个定向分支的一部分，因此从 $\Omega_{\pi\rho}$ 中消失。每个来自 $\mathcal{U}_\pi\backslash PATH(L,M)$ 的分支在 $\pi\rho$ 中保持无定向。

分支 $LCA(L,M)$ 保持无定向，当且仅当 $LCA(L,M)=\overline{LCA}(L,M)$。每个被来自 $<_\pi$ 中的 $PATH(L,M)$ 的顶点覆盖的分支会在 $<_{\pi\rho}$ 中被 $\overline{LCA}(L,M)$ 覆盖。 ■

对障碍 U 和 W，记作 $U<W$，如果 U 的最右端位置小于 W 的最右端位置，即 $U_{\max}<W_{\max}$。将 π 的障碍按它们最右端位置进行递增排序

$$U(1)<\cdots<U(l)\equiv L<\cdots<U(m)\equiv M<\cdots<U[h(\pi)]$$

并定义障碍集合

$$BETWEEN(L,M)=\{U(i):l<i<m\}$$

和

$$OUTSIDE(L,M)=\{U(i):i\notin[l,m]\}$$

引理 10.19 记 ρ 是在 π 中合并障碍 L 和 M 的反序。如果障碍集合 $BETWEEN(L,M)$ 和 $OUTSIDE(L,M)$ 都非空，那么 ρ 是安全的。

证明 $U'\in BETWEEN(L,M),U''\in OUTSIDE(L,M)$。由引理 10.18 可知反序 ρ 从 Ω_π 中删除障碍 L 和 M。这还有一个危险：通过把 K 从 π 中的非障碍变换成 $\pi\rho$ 中的障碍，ρ 可能在 $\pi\rho$ 中增加一个新的障碍 K。如果这种情况发生，在 π 中 K 不分离 L 和 M（否则，根据引理 10.18，K 会从 $\pi\rho$ 中被删除）。不失一般性，我们假定 $L<U'<M$。

如果 K 是 $\pi\rho$ 中的最小障碍，那么或者 $L<_\pi K$，或者 $M<_\pi K$（否则，K 将是 π 中的障碍）。因为 K 在 π 中不分离 L 和 M，所以 $L<_\pi K$ 及 $M<_\pi K$。因为 U' 夹在 L，M 之间，所以 $U'<_\pi K$。因而，$U'<_{\pi\rho}K$，这与 K 在 $\pi\rho$ 中的最小性矛盾。

如果 K 是 $\pi\rho$ 中的最大障碍，那么或者 $L,M\not<_\pi K$，或者 $L,M<_\pi K$（如果 $L\not<_\pi K$ 并且 $M<_\pi K$，根据引理 10.18，K 会从 $\pi\rho$ 中被删除）。如果 $L,M\not<_\pi K$，那么 $L<U'<_\pi K<M$，即 K 夹在 L 和 M 之间。因此 U'' 在 π 中位于 K 之外且 $U''\not<_{\pi\rho}K$，矛盾。如果 $L,M<_\pi K$，那么，由于 K 在 π 中是非障碍，K 从另一个障碍 N 中分离 L 和 M。因此 K 从 N 中分离出 U'。因为 N 和 U' 都在 $\pi\rho$ 中"幸存"，所以 K 在 $\pi\rho$ 中分离 N 和 U'，矛盾。

因此，ρ 从 Ω_π 中删除障碍 L 和 M，同时不在 $\pi\rho$ 中增加新的障碍，这就暗示 $\Delta h=-2$。因为 $b(\pi\rho)=b(\pi),c(\pi\rho)=c(\pi)-1,\Delta(b-c+h)=-1$，同时反序 ρ 是安全的。 ■

引理 10.20 如果 $h(\pi)>3$，那么在 π 中存在一个合并两个障碍的安全反序。

证明 将 π 的 $h(\pi)$ 个障碍按照最右端的位置进行递增排序，并设 L 和 M 是在这个次序中第一个和第 $\left(1+\frac{h(\pi)}{2}\right)$ 个障碍。因为 $h(\pi)>3$，$BETWEEN(L,M)$ 和 $OUTSIDE(L,M)$ 都是非空集，又根据引理 10.19，合并 L 和 M 的反序 ρ 是安全的。 ■

引理 10.21 如果 $h(\pi)=2$，那么在 π 中存在一个合并两个障碍的安全反序。如果 $h(\pi)=1$，那么在 π 中存在一个切割惟一障碍的安全反序。

证明 如果 $h(\pi)=2$，那么 Ω_π 不是路径图就是包含了在 π 中分离两个障碍的最大分支。在这两种情况下，在 π 中合并障碍是安全反序（引理 10.18）。如果 $h(\pi)=1$，那么引理 10.16 给出一个在 π 中切割惟一障碍的安全反序。 ■

前面的这些引理指出，障碍合并提供了一种在缺乏简单障碍的情形下寻找安全反序的方法。持否定观点的人则认为，障碍合并并不提供一个将超障碍变换成简单障碍的方法。

引理 10.22 设 ρ 是 π 中合并两个障碍 L 和 M 的反序。那么 π 中的每个超障碍（不同于 L 和 M）在 $\pi\rho$ 中仍然是超障碍。

证明 设 U 是 π 中的一个保护非障碍 U' 的超障碍（不同于 L 和 M）。显然，如果 U' 是 $\mathcal{U}_\pi\setminus U$ 中的最小障碍，那么 U 在 $\pi\rho$ 中仍然是超障碍。如果 U' 是 $\mathcal{U}_\pi\setminus U$ 中的最大障碍，那么 U' 不分离 $\mathcal{U}_\pi\setminus U$ 中的任何障碍。因此 U' 不属于 $PATH(L,M)$，且因此在 $\pi\rho$ 中"幸存"（引理 10.18）。这表明 U' 在 $\pi\rho$ 中依旧被 U 保护。 ■

10.10 反序距离的对偶定理

引理 10.20 和引理 10.21 表明除非 Ω_π 是 3-星形（3-star）图（在同一顶点上发出三条边的图）的同胚象（homeomorph），那么在 π 中存在一个安全反序。另一方面，如果在 π 中至少一个障碍是简单的，那么引理 10.17 表明 π 中存在一个安全反序。因此，安全反序可能不存在的惟一情形是当 Ω_π 是有 3 个超障碍的 3-星形图的同胚象的时候，它被称为 3-堡垒（3-fortress）[见图 10.16（b）]。

引理 10.23 如果 ρ 是破坏一个 3-堡垒 π 的反序（即 $\pi\rho$ 不是 3-堡垒），那么 ρ 是不安全的。

证明 排列 π 上的每个反序可以将 $h(\pi)$ 最多减少 2，而恰使障碍数目减少 2 的惟一操作是障碍的合并。另一方面，引理 10.18 表明在 3-堡垒中障碍的合并最多能使 $h(\pi)$ 减少 1。因此 $\Delta h\geqslant -1$。注意到对于每个不作用于同一个圈的边的反序，$\Delta c=-1$，因此，每个不作用于 3-堡垒中同一个圈的边的反序是不安全的。

如果 ρ 作用于 3-堡垒的一个无定向分支的圈上，那么它不减少障碍的数目。因为对无定向分支的圈上的一个反序，有 $\Delta c=0$，所以 ρ 是不安全的。

如果 ρ 作用于 3-堡垒的一个定向分支的圈上，那么它在 π 中不破坏任何无定向分支并且不减少障碍的数目。如果 ρ 增加障碍的数目，那么 $\Delta h\geqslant 1$ 和 $\Delta c\leqslant 1$ 表明 ρ 是不安全的。如果 $\pi\rho$ 中的障碍数目保持不变，那么 π 中的每个超障碍在 $\pi\rho$ 中依旧是超障碍，这就表明 $\pi\rho$ 是 3-堡垒，矛盾。 ■

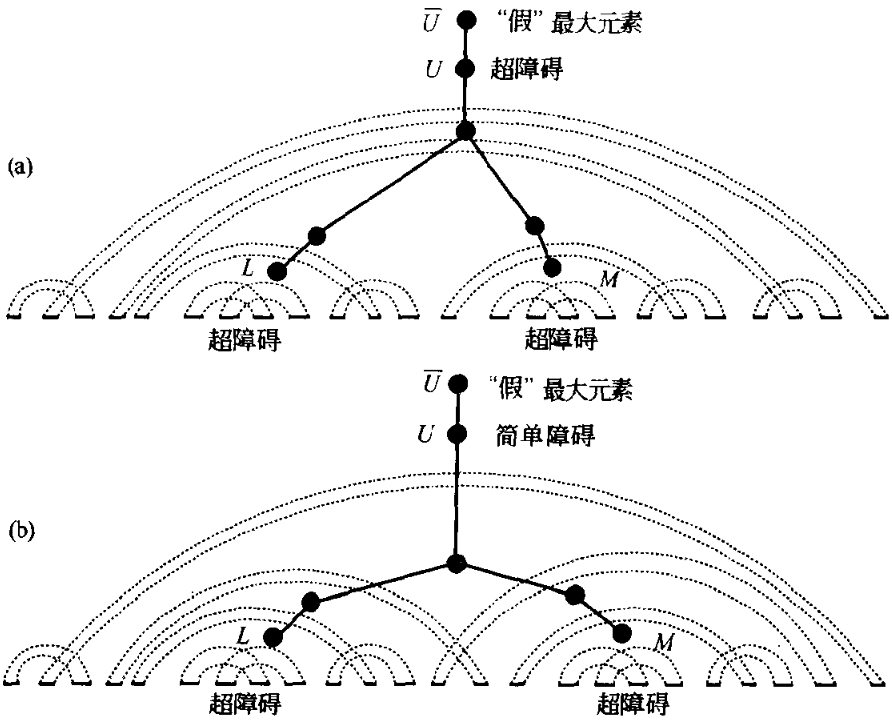


图 10.16 (a) 中的排列是一个 3-堡垒，而 (b) 中的排列不是一个堡垒，虽然带有相同的覆盖图（障碍 U 不是一个超障碍）

引理 10.24 如果 π 是一个 3-堡垒，那么 $d(\pi)=n+1-c(\pi)+h(\pi)+1$。

证明 引理 10.23 表明每个 3-堡垒的排序包含至少一个不安全反序。因此 $d(\pi)\geqslant b(\pi)-c(\pi)+h(\pi)+1$。

如果 π 有定向圈，则用定向分支中的安全填充（定理 10.7）和安全反序（定理 10.9）可以消除 π 中的所有定向分支而不影响无定向分支。

如果 π 是没有定向圈的 3-堡垒，那么在 π 中合并任意障碍的（不安全）反序 ρ 产生拥有两个障碍的排列 $\pi\rho$（引理 10.18）。再一次，用定向分支中的安全填充和安全反序可以在这种合并之后，把 $\pi\rho$ 中出现的定向圈消除（定理 10.7 和定理 10.9），这产生一个 $h(\sigma)=2$ 的排列 σ。定理 10.7、定理 10.9 和引理 10.21 表明 σ 可以通过安全填充和安全反序进行排序。这样，存在一个 π 的广义排序，使得除一个以外的所有填充和所有反序在这个排序中是安全的。因此，这个广义排序包含 $n+1-c(\pi)+h(\pi)+1$ 个反序。引理 10.11 表明，通过 $d(\pi)=n+1-c(\pi)+h(\pi)+1$ 次反序，π 的广义排序模拟了 π 的最佳（真）排序。 ■

接下来，试图在按反序排序的过程中避免造成 3-堡垒。如果成功地完成这个任务，则排列 π 在 $n+1-c(\pi)+h(\pi)$ 次反序中可以被排序。否则，指出如何在 $n+1-c(\pi)+h(\pi)+1$ 次反序中对 π 进行排序，并证明这是对排列进行排序所需的最少反序次数。如果排列 π 有奇数个障碍并且所有这些障碍都是超障碍，则

称排列 π 为堡垒。

引理 10.25 如果 ρ 是消除拥有 $h(\pi)$ 个超障碍的堡垒 π 的反序（即 $\pi\rho$ 不是拥有 $h(\pi)$ 个超障碍的堡垒），那么或者 ρ 是不安全的，或者 $\pi\rho$ 是有 $h(\pi)-2$ 个超障碍的堡垒。

证明 每个作用于一个排列上的反序能够使障碍的数目最多减少 2，惟一能够使障碍数目减少 2 的操作是障碍的合并。类似引理 10.23 的论证可知如果 ρ 不合并障碍，则 ρ 是不安全的。如果安全反序 ρ 在合并（超）π 中的障碍 L 和 M，那么引理 10.18 表明这个反序使障碍数目减少 2，并且，如果 $h(\pi)>3$，不造成新的障碍。同样，引理 10.22 表明 π 中除 L 和 M 以外的每个超障碍在 $\pi\rho$ 中仍为超障碍，这就表明 $\pi\rho$ 是有 $h(\pi)-2$ 超障碍的堡垒。 ■

引理 10.26 如果 π 是堡垒，那么 $d(\pi)\geqslant n+1-c(\pi)+h(\pi)+1$。

证明 引理 10.25 表明 π 的每个排序包含一个不安全反序或者通过将一个有 h 个（超）障碍的堡垒变换成有 $h-2$ 个（超）障碍的堡垒使得 π 的超障碍数目逐渐减少。因此，如果 π 的一个排序仅使用安全反序，那么它最终会产生一个 3-堡垒。因此，根据引理 10.23，一个堡垒的每个排序至少包含一个不安全反序，从而 $d(\pi)\geqslant n+1-c(\pi)+h(\pi)+1$。 ■

最后，用公式表示按反序排序符号排列的对偶定理。

定理 10.10 （Hannenhalli 和 Pevzner[154]）对每一个排列 π，有

$$d(\pi)=\begin{cases} n+1-c(\pi)+h(\pi)+1, & \text{如果 } \pi \text{ 是一个堡垒} \\ n+1-c(\pi)+h(\pi) & \text{其他} \end{cases}$$

证明 如果 π 有偶数个障碍，那么安全填充（定理 10.7），定向分支中的安全反序（定理 10.9）和安全障碍合并（引理 10.20 和引理 10.21）通过 $n+1-c(\pi)+h(\pi)$ 次反序产生 π 的广义排序。

如果 π 有奇数个障碍，其中至少有一个是简单的，那么存在一个安全反序切割这个简单障碍（引理 10.17）。这个安全反序产生有偶数个障碍的排列。因此，类似与先前的情形，存在一个 π 的广义排序，仅使用安全填充和 $n+1-c(\pi)+h(\pi)$ 次安全反序。

因此，如果 π 不是一个堡垒，则存在 π 的一个利用 $n+1-c(\pi)+h(\pi)$ 次反序的广义排序。引理 10.11 表明这个广义排序模拟 π 的最佳（真实的）排序。

如果 π 是堡垒，存在一系列的安全填充（定理 10.7）、有向分支中的安全反序（定理 10.9）和安全障碍合并（引理 10.20）产生一个 3-堡垒，可以被最多含有一个不安全反序的一系列反序排序。因此，存在使用 $n+1-c(\pi)+h(\pi)+1$ 次反序的 π 的广义排序。引理 10.26 表明这个广义排序模拟 π 的有 $d(\pi)=n+1-c(\pi)+h(\pi)+1$ 次反序的最佳（真实的）排序。 ■

这个定理解释了依照反序排序有符号的排列的逼近算法的惊异性能的神秘之

处。对这种性能的一个简单解释是界限 $d(\pi) \geqslant n+1-c(\pi)$ 的要求非常严格，因为 $h(\pi)$ 对于“随机”排列而言太小了。

10.11　反序排序算法

引理 10.11、引理 10.20、引理 10.17 和引理 10.21 以及定理 10.7、定理 10.9 和定理 10.10 促成了算法 Reversal-Sort，它可以最优地排序有符号的排列。

```
Reversal-Sort(π)
while π 没被排序
if π 有一个长圈
选择一个 π 的安全 (g, b)-填充 ρ（定理 10.7）
else if π 有一个定向分支
在此分支中选择一个安全反序 ρ（定理 10.9）
else if π 有偶数个障碍
选择一个合并 π 中两个障碍的安全反序 ρ（引理 10.20、引理 10.21）
else if π 至少有一个简单障碍
选择一个在 π 中切割这个障碍的安全反序 ρ（引理 10.17、引理 10.21）
else if π 是有多于 3 个超障碍的堡垒
选择一个在 π 中合并两个（超）障碍的安全反序 ρ（引理 10.20）
else /* π 是一个 3-堡垒 */
选择一个在 π 中合并两个任意（超）障碍的（不）安全反序 ρ
π←π·ρ
endwhile
通过计算 π 的广义排序模拟（真实的）π 的排序（引理 10.11）
```

定理 10.11　Reversal-Sort(π) 对 n 阶排列 π 的最佳排序耗时 $O(n^4)$。

证明　定理 10.10 表明 Reversal-Sort 通过一系列反序和包含 $d(\pi)$ 个反序的填充提供了一个 π 广义排序。引理 10.11 表明这个广义排序通过 $d(\pi)$ 个反序模拟最佳（真实的）π 的排序。

注意，Reversal-Sort 中 ***while*** 循环的每次迭代将 $complexity(\pi)+3d(\pi)$ 的量至少减少 1，这表明 Reversal-Sort 的迭代次数不超过 $4n$。最“昂贵的”迭代是在定向分支中搜索一个安全反序。因为对简单排列它可以在 $O(n^3)$ 时间内被执行，所以 Reversal-Sort 的总运行时间是 $O(n^4)$。■

下面我们讨论 Reversal-Sort 的一个简化形式，它不使用填充且运行时间是 $O(n^5)$。定义

$$f(\pi)=\begin{cases}1, & \text{如果 } \pi \text{ 是一个堡垒}\\ 0, & \text{其他}\end{cases}$$

如果 $\Delta(c-h-f)=-1$，则反序 ρ 是有效的。定理 10.6 和引理 10.26 的证明表明 $\Delta(c-h-f)\geqslant-1$。由此结果和定理 10.10 可推出下述定理。

定理 10.12 对每个排列 π，在 π 中存在一个有效反序。每个有效反序的序列对 π 进行最佳排序。

定理 10.12 促成的 Reversal-Sort 算法的这个简化版本在实际使用中是非常快速的。

Reversal-Sort-Simple(π)

1. **while** π 没被排序
2. 在 π 中选择一个有效反序 ρ （定理 10.12）
3. $\pi\leftarrow\pi\cdot\rho$
4. **endwhile**

10.12 人基因组变换成老鼠基因组

多染色体基因组重排的分析利用了单基因组的对偶定理（Hannenhalli 和 Pevzner[154]）和两个附加的思想，称为染色体翻转（flipping）和加帽（capping）。在对多染色体基因组进行基因重排的研究中，染色体定义为基因的序列，而基因组定义为染色体的集合。设 $\Pi=\{\pi(1),\cdots,\pi(N)\}$ 是包含 N 条染色体的基因组，设 $\pi(i)=\pi(i)_1\cdots\pi(i)_{n_i}$，$n_i$ 是第 i 个染色体中基因的个数。每个染色体 π 既可以从左向右（即 $\pi=\pi_1\cdots\pi_n$）看，又可以从右向左（即 $-\pi=-\pi_n\cdots-\pi_1$）看，这就产生同一染色体的两个等价表现形式。从这个观点出发，一个含有 3 条染色体的基因组 $\{\pi(1),\pi(2),\pi(3)\}$ 等价于 $\{\pi(1),-\pi(2),\pi(3)\}$ 或 $\{-\pi(1),\pi(2),-\pi(3)\}$，即染色体的方向是无关的。在多染色体基因组中，4 个最普通的基本重排事件是反序、易位、融合和裂变。

区分不包括染色体的尾部的内部反序（internal reversal）[即一个含有 n 个基因的染色体 π 的反序 $\rho(\pi, i, j)$，$1<i\leqslant j<n$]，以及包括染色体的尾部的前缀反序（prefix reversal）（即 $i=1$ 或 $j=n$）。如果一个易位既不是融合又不是裂变，那它是内部的（internal）。

对染色体 $\pi=\pi_1\cdots\pi_n$，数字 $+\pi_1$ 和 $-\pi_n$ 称为 π 的尾部。注意，改变染色体的方向并不改变它的尾部的集合。含有 N 个染色体的基因组 Π 的尾部由包含 $2N$ 个元素的集合 $\mho(\Pi)$ 组成。在这一节中，考虑 $\mho(\Pi)=\mho(\Gamma)$ 的共尾部（co-tailed）基因组 Π 和 Γ。对共尾部基因组，内部反序和易位对基因组排序是足够的——即前缀反序，融合和裂变可予忽略（这种假设的有效性会在以后加以说明）。对染色体 $\pi=\pi_1\cdots\pi_n$ 和 $\sigma=\sigma_1\cdots\sigma_m$，融合 $\pi_1\cdots\pi_n\sigma_1\cdots\sigma_m$ 记为 $\pi+\sigma$，融合 $(\pi_1\cdots\pi_n-\sigma_m\cdots-\sigma_1)$ 记为 $\pi-\sigma$。给定基因组 Π 的染色体 $[\pi(1),\cdots,\pi(N)]$ 的排序和翻转向量 $s=[s(1),\cdots,s(N)]$，$s(i)\in(-1,+1)$，能够形成 Π 的一个连接，它

是 $\sum_{i=1}^{N} n_i$ 个元素的排列 $\Pi(s)=s(1)\pi(1)+\cdots+s(N)\pi(N)$。根据翻转向量的选取，对 Π 中染色体的 $N!$ 个排序，每一个都存在 2^N 个 Π 的多联体。如果基因组 Π 中的染色体的次序是固定的，称 Π 为有序基因组。

不失一般性，在这节中假定 $\Gamma=(\gamma_1，\cdots，\gamma_N)$ 是（有序）基因组，并且 $\gamma=\gamma_1+\cdots+\gamma_N$ 是恒等排列。记作 $d(\Pi)\equiv d(\Pi,\Gamma)$，并且把基因组 Π 排序成 Γ 的问题简称为基因组 Π 的排序。

使用下述思想分析共尾部基因组。给定基因组 Π 的一个多联体 π，可以通过反序最佳地排序 π。在排序中每个反序对应于基因组 Π 的（不必是最佳的）排序中的一个反序或者一个易位。例如，作用于染色体 $\pi=\pi_1\cdots\pi_n$ 和 $\sigma=\sigma_1\cdots\sigma_m$ 上的易位 $\rho(\pi，\sigma，i，j)$，图 10.17 也可视为作用于 $\pi-\sigma$ 上的反序 $\rho[\pi-\sigma,i,n+(m-j+1)]$（反之亦然）。定义 Π 的一个最佳连接为 Π 的所有多联体中具有最小反序距离 $d(\pi)$ 的一个多联体 π。下面证明 Π 的一个最佳连接的排序模拟了基因组 Π 的一个最佳排序。此方法把 Π 的排序问题简化为寻找 Π 的最佳连接问题。

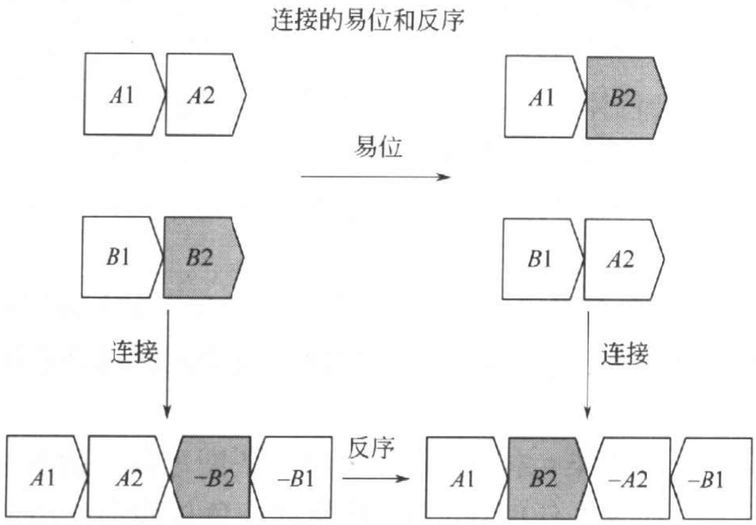


图 10.17 在一个连接的基因组中易位可用反序模拟

下面，根据圈的数目，预估长度大于 2 的圈的数目，即 $c(\pi)-a(\pi)$，其中 $a(\pi)$ 是邻接数（每个邻接是长度为 2 的圈）。因为 $b(\pi)=n+1-a(\pi)$，Hannenhalli-Pevzner 定理可以重写为 $d(\pi)=b(\pi)-c(\pi)+h(\pi)+f(\pi)$，其中 $f(\pi)=1$，如果 π 是堡垒；否则 $f(\pi)=0$。

设 π 是 $\Pi=(\pi(1),\cdots,\pi(N))$ 的一个多联体。每个 $\pi(i)$ 的尾部对应于断点图 $G(\pi)$ 上的两个顶点，其中一个恰是在多联体 π 中的染色体 $\pi(i)$ 的顶点之中的边界（最左或最右）顶点。用术语尾部扩展来表示在 $G(\pi)$ 中的这种顶点。多联体 π 的断点图 $G(\pi)$ 中的边是染色体间（interchromosomal）的，如果它连接 Π

的不同染色体的顶点，否则是染色体内（intrachromosomal）的。π 的分支是染色体间的，如果它包含染色体间的边，否则是染色体内的。

$G(\pi)$ 中的每一条染色体间的黑边连接两个尾部。设 $b_{tail}(\Pi)$ ［注意到 $b_{tail}(\Pi)=N-1$］是 $G(\pi)$ 中染色体间的黑边的数目。注意共尾部基因组，$G(\Pi)$ 中的尾部仅与尾部相邻，因此包含一个尾部的圈仅有一个尾部。设 $c_{tail}(\Pi)$ 是 $G(\pi)$ 中包含尾部的圈的数目。定义 $b(\Pi)=b(\pi)-b_{tail}(\pi)$ ［注意到 $b(\Pi)=n-N$ 及 $b(\pi)=n-1$］，$c(\Pi)=c(\pi)-c_{tail}(\pi)$。

考虑 π 中染色体内的无定向分支 $\mathcal{IU}_\pi$ 的集合。对 $\mathcal{IU}_\pi$ 集合来说，障碍，超障碍和堡垒分别称为结点（knot），超结点（superknot）和堡垒-结点（fortress-of-knot）。设 $k(\Pi)$ 是 Π 的多联体 π 中的结点数。如果 π 是堡垒-结点，定义 $f(\Pi)=1$，否则 $f(\Pi)=0$。显然，$b(\Pi)$、$c(\Pi)$、$k(\Pi)$ 和 $f(\Pi)$ 都不依赖于多联体 π 的选取。

引理 10.27 对共尾部基因组 Π 和 Γ，$d(\Pi)\geq b(\Pi)-c(\Pi)+k(\Pi)+f(\Pi)$。

证明 参看引理 10.26 的证明。 ■

一个（有序）基因组 Π 的多联体 $\Pi(s)$ 和 $\Pi(s')$ 是 i-成对的（i-twins），如果在 $\Pi(s)$ 和 $\Pi(s')$ 中所有染色体的方向除第 i 条外一致，即 $s(i)=-s'(i)$，$s(j)=s'(j)$，$j\neq i$。染色体 $\pi(i)$ 在 $\Pi(s)$ 中是完全翻转（properly flipped）的，如果从这条染色体出发的全部染色体间的边属于 $\Pi(s)$ 中的定向分支。多联体 π 是完全翻转的，如果 π 中的每条染色体是完全翻转。下面的引理证明完全翻转多联体的存在性。

引理 10.28 如果染色体 $\pi(i)$ 在 $\pi=\Pi(s)$ 中不是完全翻转的，那么它在 π 的 i-成对的 π' 中是完全翻转的。此外，π 的每个完全翻转的染色体在 π' 中保持完全翻转。

证明 设 g 是 π 中由染色体 $\pi(i)$ 出发的一条染色体间的灰边，且属于 π 中的一个无定向分支。注意，任何在 $\pi(i)$ 出发的染色体间的灰边的方向在 π 和 π' 中是不同（即 π 的一个无定向边在 π' 中成为定向边，反之亦然）。因为所有在 π 中与 g 交叉的边是无定向的，所以由 $\pi(i)$ 出发的并在 π 中与 g 交叉的每条染色体间的边在 π' 中是定向的。

全部由 $\pi(i)$ 出发的染色体间的边在 π 中不与 g 交叉，但在 π' 中与 g 交叉。由于 g 在 π' 中定向，全部这种边属于 π' 中一个包含 g 的定向分支。因此，$\pi(i)$ 在 π' 中完全翻转。

设 $\pi(j)$ 是在 π 中的完全翻转染色体。如果 $\pi(j)$ 在 π' 中不是完全翻转的，那在 π' 中存在一个染色体间的无定向分支，含有一条由 $\pi(j)$ 出发的染色体间的灰边。如果 U 在 π' 中没有一条由 $\pi(i)$ 出发的边，那么 U 在 π 中是无定向分支，这表明 $\pi(j)$ 在 π 中不是完全翻转的，矛盾。如果 U 有一条由 $\pi(i)$ 出发的（无

向）灰边 h，显而易见，这条边在 π' 中不与 g 交叉。因此，h 在 π 中与 g 交叉且有向，这样表明 g 在 π 中属于一个有向分支，矛盾。 ■

引理 10.28 证明了完全翻转多联体 $\pi=\Pi(s)$ 的存在性，有 $h(\pi)=k(\Pi)$ 和 $f(\pi)=f(\Pi)$。下面指出存在一个按 $b(\pi)-c(\pi)+h(\pi)+f(\pi)$ 次反序的 π 的排序模拟按 $b(\Pi)-c(\Pi)+k(\Pi)+f(\Pi)$ 次内部反序和易位的 Π 的排序。

定理 10.13 （Hannenhalli 和 Pevzner[153]）对于共尾部基因组 Π 和 Γ，$d(\Pi,\Gamma)\equiv d(\Pi)=b(\Pi)-c(\Pi)+k(\Pi)+f(\Pi)$。

证明 反证之，设 Π 是在全部基因组之中的 $b(\Pi)-c(\Pi)+h(\Pi)+f(\Pi)$ 值最小的基因组。又设 π 是 Π 的一个完全翻转多联体，其 $b_{tail}(\pi)-c_{tail}(\pi)$ 值在 Π 的所有完全翻转多联体之中最小。

如果 $b_{tail}(\pi)=c_{tail}(\pi)$ （即每条染色体间的黑边被包括在一个长度为 2 的圈中），那么存在 π 的一个最佳排序，它通过对染色体内的黑边作用 $b(\pi)-c(\pi)+k(\pi)+f(\pi)$ 次反序而得到（Hannenhalli 和 Pevzner[154]）。每个这样的反序 ρ 能够被模拟为 Π 上的一个内反序或者一个内易位，于是通过 $b(\pi)-c(\pi)+k(\pi)+f(\pi)$ 次内反序/易位，就产生 Π 的一个排序。因为 π 是一个完全翻转多联体，所以 $b(\pi)=b(\Pi)+b_{tail}(\pi)$，$c(\pi)=c(\Pi)+c_{tail}(\pi)$，$h(\pi)=k(\Pi)$ 和 $f(\pi)=f(\Pi)$。因此，π 的最佳排序模拟了通过 $b(\Pi)-c(\Pi)+k(\Pi)+f(\Pi)$ 次内反序/易位而得到的 Π 的最佳排序。

如果 $b_{tail}(\pi)>c_{tail}(\pi)$，那么在长度大于 2 的圈里存在一条染色体间的黑边，且这条边属于 π 的一个定向分支（因为在一个完全翻转多联体中，每条交互染色体的黑边属于一个定向分支）。Hannenhalli 和 Pevzner[154] 证明了如果在 π 中存在一个有向分支，那么在 π 中存在一个作用于这个分支中的一个有向圈的黑边上的反序 ρ，使得 $c(\pi\rho)=c(\pi)+1$。此外，这个反序在 $\pi\rho$ 中不产生新的无定向分支，并且 $h(\pi\rho)=h(\pi)$、$f(\pi\rho)=f(\pi)$。注意到每个包含染色体尾部的圈属于 π 的一个定向分支，且由尾部之间全部的边组成。因此，ρ 或者作用在两个染色体内的黑边上，或者作用在两条属于这个分支的某个定向圈的染色体间的黑边上。

作用在两条染色体间的黑边上的反序 ρ 可以被看作是把

$$\Pi=(\pi(1),\cdots,\pi(i-1),\pi(i),\cdots,\pi(j),\pi(j+1),\cdots,\pi(N))$$

的多联体 π 变换成多联体 $\pi\rho'=\Pi'(s')$，其中 Π' 是该染色体新的排序

$$(\pi(1),\cdots,\pi(i-1),\pi(j),\cdots,\pi(i),\pi(j+1),\cdots,\pi(N))$$

并且

$$s'=(s(1),\cdots,s(i-1),-s(j),\cdots,-s(i),s(j+1),\cdots,s(N))$$

因此，$b_{tail}(\pi\rho)-c_{tail}(\pi\rho)=b_{tail}(\pi)-[c_{tail}(\pi)+1]$ 且 $\pi\rho$ 是 Π 的完全翻转多联体，这与 $b_{tail}(\pi)-c_{tail}(\pi)$ 最小相矛盾。

如果反序 ρ 作用于两条染色体内的黑边，那么 $\pi\rho$ 是 $\Pi\rho$ 的一个完全翻转多联

体，这表明

$$\begin{aligned} b(\Pi)-c(\Pi)+k(\Pi)+f(\Pi) &= [b(\pi)-b_{\text{tail}}(\pi)]-[c(\pi)-c_{\text{tail}}(\pi)]+h(\pi)+f(\pi) \\ &= [b(\pi\rho)-b_{\text{tail}}(\pi\rho)]-[c(\pi\rho)-1-c_{\text{tail}}(\pi\rho)]+ \\ &\quad h(\pi\rho)+f(\pi\rho) \\ &= b(\Pi\rho)-c(\Pi\rho)+h(\Pi\rho)+f(\Pi\rho)+1 \end{aligned}$$

因为 $b(\Pi)-c(\Pi)+k(\Pi)+f(\Pi)>b(\Pi\rho)-c(\Pi\rho)+h(\Pi\rho)+f(\Pi\rho)$，故对于基因组 $\Pi\rho$ 定理成立。因此，$d(\Pi)\leqslant d(\Pi\rho)+1=b(\Pi)-c(\Pi)+k(\Pi)+f(\Pi)$。 ■

10.13　加帽染色体

现在回到通常的情况，即基因组 Π 和 Γ 可能有不同的尾部集和不同数目的染色体。下面描述一个计算 $d(\Pi,\Gamma)$ 的算法，其计算复杂度与基因数目成多项式关系，但与染色体数目成指数关系。这个算法导出一个（真正的）多项式时间算法，下面几节将专门予以讨论。

设 Π 和 Γ 是两个分别含有 M 和 N 条染色体的基因组。不失一般性，假定 $M\leqslant N$，并用 $N-M$ 个空染色体扩充 Π（见图 10.18）。从而 $\Pi=\{\pi(1),\cdots,\pi(N)\}$ 和 $\Gamma=\{\gamma(1),\cdots,\gamma(N)\}$ 包含同样数目的染色体。设 $\{cap_0,\ \cdots,\ cap_{2N-1}\}$ 是 $2N$ 个不同正整数（称为帽子）的集合，它们与来自 Π 的基因不同（对 Γ 也等价）。设 $\hat{\Pi}=\{\hat{\pi}(1),\cdots,\hat{\pi}(N)\}$ 是通过在 Π 中的每条染色体的末端加上帽子而得到的基因组，即

$$\hat{\pi}(i)=cap_{2(i-1)},\pi(i)_1,\cdots,\pi(i)_{n_i},cap_{2(i-1)+1}$$

注意，Π 里的每个反序/易位对应 $\hat{\Pi}$ 里的一个内反序/易位。如果这个易位是裂变，则假定在 Π 里有足够的空染色体（此假定的有效性稍后会得到证明）。

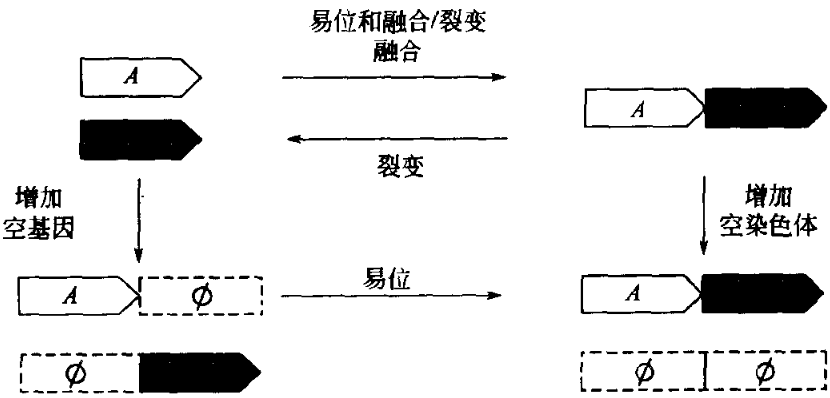


图 10.18　通过引入空染色体的易位可以模拟融合/裂变

每个 Π 到 Γ 的排序导致 $\hat{\Pi}$ 到基因组

$$\hat{\Gamma}=\{\hat{\gamma}(1),\cdots,\hat{\gamma}(N)\}$$

（称为 Γ 的加帽）的排序，其中

$$\hat{\gamma}(i)=[(-1)^{j}cap_{j},\hat{\gamma}(i)_{1},\cdots,\hat{\gamma}(i)_{m_{i}},(-1)^{k+1}cap_{k}],0\leqslant j,k\leqslant 2N-1$$

基因组 $\hat{\Pi}$ 和 $\hat{\Gamma}$ 是共尾部的，因为 $\mho(\hat{\Pi})=\mho(\hat{\Gamma})=\bigcup_{i=0}^{2N-1}(-1)^{i}cap_{i}$。存在 $(2N)!$ 个不同的 Γ 的加帽，每个加帽根据 $\hat{\Gamma}$ 中 $\hat{\Pi}$ 的 $2N$ 个帽子的分布而定义。把 Γ 的 $(2N)!$ 个加帽的集合记为 Γ。下面的引理产生一个计算基因组的距离的算法，它和基因的数目成多项式关系，而和染色体 N 的数目成指数关系。

引理 10.29　$d(\Pi,\Gamma)=\min\limits_{\hat{\Gamma}\in\Gamma} b(\hat{\Pi},\hat{\Gamma})-c(\hat{\Pi},\hat{\Gamma})+k(\hat{\Pi},\hat{\Gamma})+f(\hat{\Pi},\hat{\Gamma})$。

证明　由定理 10.13 以及观察到的通过内反序/易位使 $\hat{\Pi}$ 变成基因组 $\hat{\Gamma}\in\Gamma$ 的每个排序导致 Π 变成 Γ 的排序可证得。■

设 $\hat{\pi}$ 和 $\hat{\gamma}$ 是（有序的）加帽 $\hat{\Pi}$ 和 $\hat{\Gamma}$ 的任意多联体。设 $G(\hat{\Pi},\hat{\Gamma})$ 是一张通过删除 $G(\hat{\pi},\hat{\gamma})$ 的基因组 $\hat{\Pi}$（或等价地对 $\hat{\Gamma}$）的所有尾部 [$G(\hat{\pi},\hat{\gamma})$ 的顶点] 而得到的图。不同的加帽 $\hat{\Gamma}$ 对应不同的图 $G(\hat{\Pi},\hat{\Gamma})$。图 $G(\hat{\Pi},\hat{\Gamma})$ 有 $2N$ 个对应帽子的顶点；和这些顶点关联的灰边完整地定义了加帽 $\hat{\Gamma}$。因此，删除这 $2N$ 条灰边就把 $G(\hat{\Pi},\hat{\Gamma})$ 变换成图 $G(\Pi,\Gamma)$，这不依赖于加帽 $\hat{\Gamma}$ [见图 10.19（a）～图 10.19（d）]。

图 $G(\Pi,\Gamma)$ 包含 $2N$ 个度为 1 的顶点，它们对应于 Π 的 $2N$ 个帽子 [称为 Π 帽子（cap）]，也包含 $2N$ 个度数为 1 的顶点，它们对应于 Γ 的 $2N$ 个尾部 [称为 Γ 尾部（tail）]。因此，$G(\Pi,\Gamma)$ 是圈和 $2N$ 条路的集合，每条路都以黑边起始和终止。一条路是 $\Pi\Pi$ 路（path）（$\Gamma\Gamma$ 路），如果它的起点和终点都是 Π 帽子（Γ 尾部）。一条路是 $\Pi\Gamma$ 路，如果它的起点是 Π 帽子而终点则是 Γ 尾部。$G(\Pi,\Gamma)$ 里的一个顶点是 $\Pi\Gamma$ 顶点（vertex），如果它是一条 $\Pi\Gamma$ 路中的 Π 帽子；$G(\Pi,\Gamma)$ 里的一个顶点是 $\Pi\Pi$ 顶点，如果它是在一条 $\Pi\Pi$ 路上的 Π 帽子。$\Gamma\Pi$ 顶点和 $\Gamma\Gamma$ 顶点被类似定义 [见图 10.19（d）]。

每个加帽 $\hat{\Gamma}$ 对应于添加 $2N$ 条灰边到图 $G(\Pi,\Gamma)$，每条边连接一个 Π 帽子和一个 Γ-尾部。这些边把 $G(\Pi,\Gamma)$ 变换成对应于一个加帽 $\hat{\Gamma}$ 的图 $G(\hat{\Pi},\hat{\Gamma})$ [见图 10.19（e）]。

定义 $b(\Pi,\Gamma)$ 为 $G(\Pi,\Gamma)$ 中黑边的数目，又定义 $c(\Pi,\Gamma)$ 为 $G(\Pi,\Gamma)$ 中圈和路的全部数目。参数 $b(\Pi,\Gamma)=b(\hat{\Pi},\hat{\Gamma})$ 不依赖于加帽 $\hat{\Gamma}$。显然，$c(\hat{\Pi},\hat{\Gamma})\leqslant c(\Pi,\Gamma)$，当 $G(\Pi,\Gamma)$ 中的每条路被 $G(\hat{\Pi},\hat{\Gamma})$ 中的灰边所“封闭”时，有 $c(\hat{\Pi},\hat{\Gamma})=c(\Pi,\Gamma)$。因为 $G(\hat{\Pi},\hat{\Gamma})$ 中的每个包含 $\Pi\Pi$ 路的圈至少有一条路，所以 $c(\hat{\Pi},\hat{\Gamma})\leqslant c(\Pi,\Gamma)-p(\Pi,\Gamma)$，其中 $\rho(\Pi,\Gamma)$ 是 $G(\Pi,$

Γ）中 ΠΠ 路（或等价地为 ΓΓ 路）的数目。

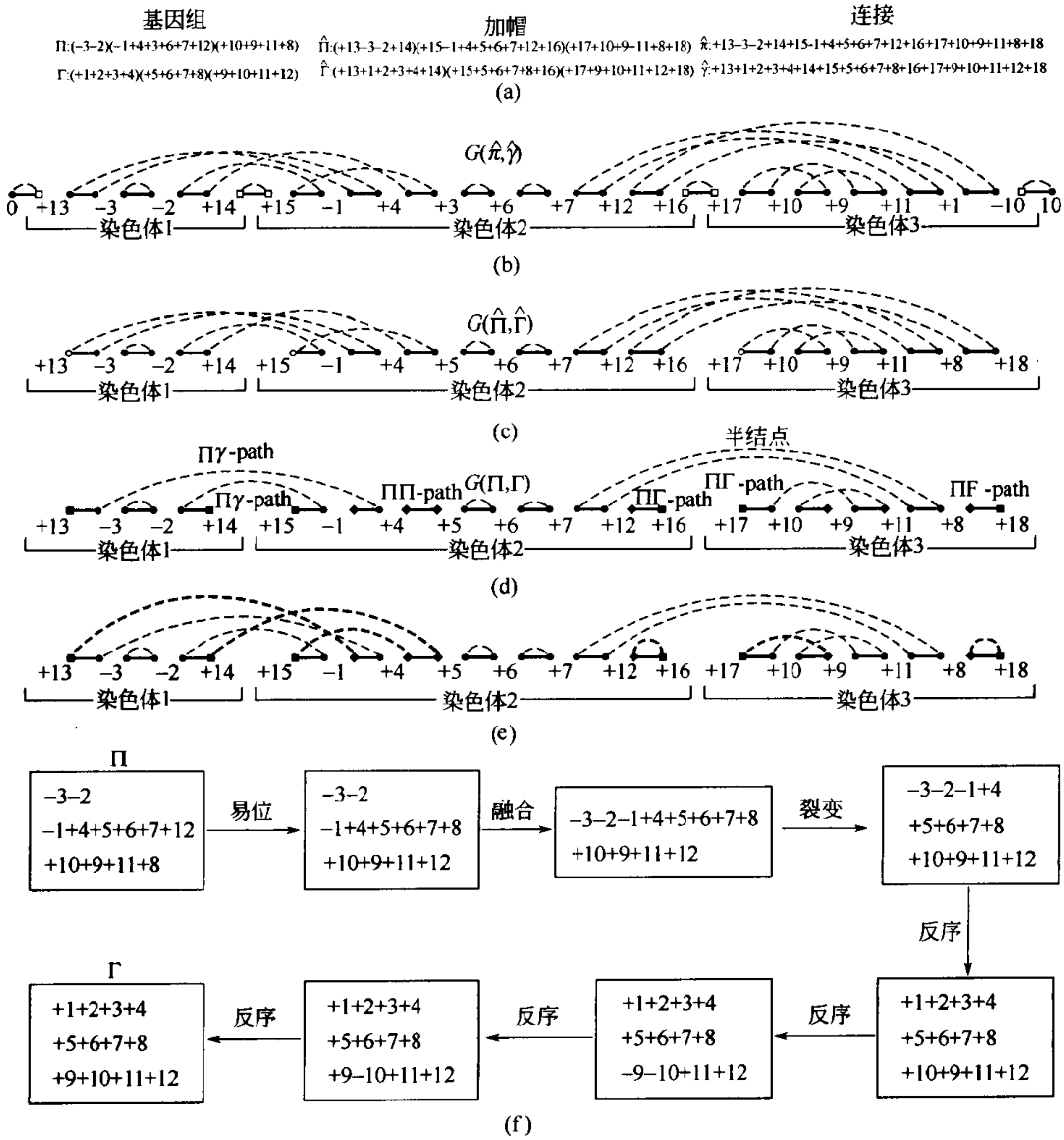


图 10.19　(a) 基因组 Π 和 Γ，加帽 $\hat{\Pi}$ 和 $\hat{\Gamma}$，连接 $\hat{\pi}$ 和 $\hat{\gamma}$；(b) 图 $G(\hat{\pi},\hat{\gamma})$ 尾部用白方框表示；(c) 通过删除尾部，从 $G(\hat{\pi},\hat{\gamma})$ 得到的图 $G(\hat{\Pi},\hat{\Gamma})$。帽子用白圆圈表示；(d) 拥有 4 个圈和 6 条路的图 $G(\Pi,\Gamma)[c(\Pi,\Gamma)=10]$。Π 帽子用方框表示，而 Γ 尾部用菱形表示。对基因组 Π 和 Γ，有 $b(\Pi,\Gamma)=15$，$r(\Pi,\Gamma)=0$，$p(\Pi,\Gamma)=1$，$s(\Pi,\Gamma)=1$，$gr(\Pi,\Gamma)=fr(\Pi,\Gamma)=0$。因此，$d(\Pi,\Gamma)=15-10+1+0+\left(\frac{1-0+0}{2}\right)=7$；(e) 对应 $\hat{\Gamma}=(+13+1+2+3+4-15)(-14+5+6+7+8+18)(+17+9+10+11+12+16)$ 的一个最佳加帽的图 $G(\hat{\Pi},\hat{\Gamma})$。增加的灰边用粗虚线表示；(f) 将 Π 转化为 Γ 的一个有 7 次重排的最佳排序

用通常的方法定义图 $G(\Pi, \Gamma)$ 中的交叉圈/路，定向/无定向分支等的概念，且对 $G(\Pi, \Gamma)$ 中的圈和路不作区别。如果 $j\in[\bar{U}_{\min}, \bar{U}_{\max}]$，则顶点 π_j 在 π 的分支 U 内。如果基因组 Π 和 Γ 的一个染色体内的分支既没有 Π 帽子又没有 Γ 尾部，则该分支称为真分支（real component）。

对基因组 Π 和 Γ，定义 $\mathcal{RU}(\Pi, \Gamma)$ 为真分支的集合，定义 $\mathcal{IU}(\Pi, \Gamma)$ 为染色体内的分支的集合［和用图 $G(\Pi, \Gamma)$ 定义一样］。显然，$\mathcal{RU}(\Pi, \Gamma)\subseteq\mathcal{IU}(\Pi, \Gamma)$。对于集合 $\mathcal{RU}(\Pi, \Gamma)$，障碍、超障碍和堡垒分别称为真结点（real-knot）、超真结点（super-real-knot）和真结点堡垒（fortress-of-real-knot）。设 RK 是真结点的集合［即集合 $\mathcal{RU}(\Pi, \Gamma)$ 的障碍］，设 K 是结点的集合［即集合 $\mathcal{IU}(\Pi, \Gamma)$ 的障碍］。一个来自集合 $K\setminus RK$ 的结点称为半结点（semi-knot），如果它内部不包含 ΠΠ 顶点或 ΓΓ 顶点。显然，每个半结点包含一条 ΠΓ 路（否则，它应当是一个真结点）。基因组 Π 和 Γ 的真结点和半结点的数目分别记为 $r(\Pi, \Gamma)$ 和 $s(\Pi, \Gamma)$。显然，$k(\hat{\Pi}, \hat{\Gamma})\geqslant r(\Pi, \Gamma)$ 这表明

$$b(\hat{\Pi},\hat{\Gamma})-c(\hat{\Pi},\hat{\Gamma})+k(\hat{\Pi},\hat{\Gamma})\leqslant b(\Pi,\Gamma)-c(\Pi,\Gamma)+p(\Pi,\Gamma)+r(\Pi,\Gamma)$$

然而，这个界限是不严格，因为它假定存在一个加帽 $\hat{\Gamma}$，同时地使得 $c(\hat{\Pi}, \hat{\Gamma})$ 取极大而 $k(\hat{\Pi}, \hat{\Gamma})$ 取极小。考虑 $s(\Pi, \Gamma)$ 的计算则可以得到一个对基因组距离来说更好的界限，即除基因组距离之外至多有 1 个重排。

10.14 帽子和尾部

基因组 Π 和 Γ 是相关的，如果 $G(\Pi, \Gamma)$ 中的全部真结点位于同一染色体上，否则是非相关的。在这一节中，仅对非相关基因组进行分析（这是因为分析相关基因组涉及一些附加的技术困难），并且证明 $d(\Pi, \Gamma)$ 的严格界限（这个界限导致一个更复杂的用于对偶定理证明的势函数）

$$b(\Pi,\Gamma)-c(\Pi,\Gamma)+p(\Pi,\Gamma)+r(\Pi,\Gamma)+\left\lceil\frac{s(\Pi,\Gamma)}{2}\right\rceil\leqslant d(\Pi,\Gamma)\leqslant$$

$$b(\Pi,\Gamma)-c(\Pi,\Gamma)+p(\Pi,\Gamma)+r(\Pi,\Gamma)+\left\lceil\frac{s(\Pi,\Gamma)}{2}\right\rceil+1$$

下面的引理提出一种方法，用定向边连接 $G(\Pi, \Gamma)$ 中的一些路。

引理 10.30 对 $G(\Pi, \Gamma)$ 中的每条 ΠΠ 路和 ΓΓ 路，存在一条染色体内的或定向的灰边，连接这些路成为一条 ΠΓ 路。

引理 10.31 对位于同一染色体上的每两条无定向 ΠΓ 路，存在一条定向灰边，连接这些路成为一条 ΠΓ 路。

在搜索最佳加帽中，首先忽略引理 10.29 中的项 $f(\hat{\Pi}, \hat{\Gamma})$，并寻找一个加帽，使其基因组距离 $d(\hat{\Pi}, \hat{\Gamma})$ 与最佳的基因组距离之间差距不大于 1。下面的

定理提出寻找这种“几乎最佳”的加帽 $\hat{\Gamma}$ 的方法。

定理 10.14

$$\min_{\hat{\Gamma}\in\Gamma} b(\hat{\Pi},\hat{\Gamma})-c(\hat{\Pi},\hat{\Gamma})+k(\hat{\Pi},\hat{\Gamma})=b(\Pi,\Gamma)-c(\Pi,\Gamma)+p(\Pi,\Gamma)+r(\Pi,\Gamma)+\left\lceil\frac{s(\Pi,\Gamma)}{2}\right\rceil$$

证明 每个加帽 $\hat{\Gamma}$ 定义一个把 $G(\Pi,\Gamma)$ 转化成 $G(\hat{\Pi},\hat{\Gamma})$ 的变换，即在 $G(\Pi,\Gamma)$上连续地增加 $2N$ 条灰边：$G(\Pi,\Gamma)=G_0\xrightarrow{g_1}G_1\xrightarrow{g_2}\cdots\xrightarrow{g_{2N}}G_{2N}=G(\hat{\Pi},\hat{\Gamma})$。对于图 G_i，参数 $b_i=b(G_i),c_i=c(G_i),p_i=p(G_i),r_i=r(G_i),s_i=s(G_i)$的定义和图 $G_0=G(\Pi,\Gamma)$ 中的定义一样。对于参数 ϕ，定义 $\Delta\phi_i=\phi_i-\phi_{i-1}$，即 $\Delta c_i=c_i-c_{i-1}$等。记 $\Delta_i=\left(c_i-p_i-r_i-\left\lceil\frac{s_i}{2}\right\rceil\right)-\left(c_{i-1}-p_{i-1}-r_{i-1}-\left\lceil\frac{s_{i-1}}{2}\right\rceil\right)$。下面证明当 $1\leqslant i\leqslant 2N$ 时，$\Delta_i\leqslant 0$，即添加灰边不提高参数 $c(\Pi,\Gamma)-p(\Pi,\Gamma)-r(\Pi,\Gamma)-\left\lceil\frac{s(\Pi,\Gamma)}{2}\right\rceil$。对固定的 i，忽略指标 i，即记 $\Delta=\Delta_i$ 等。

取决于边 g_i，下面的情形是可能的（下面的分析假定 Π 和 Γ 是不相关的）。

情形 1 边 g_i“封闭”一条 $\Pi\Gamma$ 路（即 g_i 连接同一 $\Pi\Gamma$ 路上的一个 $\Pi\Gamma$ 顶点和一个 $\Gamma\Pi$ 顶点）。如果这个顶点是半结点中惟一的 $\Pi\Gamma$ 顶点，那么 $\Delta c=0$，$\Delta p=0$，$\Delta r=1$，$\Delta s=-1$（注意到这对于相关的基因组可能不成立）。否则 $\Delta c=0$，$\Delta p=0$，$\Delta r=0$，$\Delta s=0$。两种情形下都有 $\Delta\leqslant 0$。

情形 2 边 g_i 连接不同 $\Pi\Gamma$ 路上的一个 $\Pi\Gamma$ 顶点和一个 $\Gamma\Pi$ 顶点。这条边至多“消除”两个半结点，且 $\Delta c=-1$，$\Delta p=0$，$\Delta r=0$ 和 $\Delta s\geqslant -2$。因此 $\Delta\leqslant 0$。

情形 3 边 g_i 连接一个 $\Pi\Gamma$ 顶点和一个 $\Gamma\Gamma$ 顶点（或者是一个 $\Gamma\Pi$ 顶点和一个 $\Pi\Pi$ 顶点）。这条边至多“消除”一个半结点，且 $\Delta c=-1$，$\Delta p=0$，$\Delta r=0$ 和 $\Delta s\geqslant -2$。这表明 $\Delta\leqslant 0$。

情形 4 边 g_i 连接一个 $\Pi\Pi$ 顶点和一个 $\Gamma\Gamma$ 顶点。这条边不消除任何半结点，且 $\Delta c=-1$，$\Delta p=-1$，$\Delta r=0$ 和 $\Delta s\geqslant 0$。这表明 $\Delta\leqslant 0$。

注意到 $b_{2N}=b(\hat{\Pi},\hat{\Gamma})=b(\Pi,\Gamma)=b_0,c_{2N}=c(\hat{\Pi},\hat{\Gamma}),p_{2N}=0,s_{2N}=0,r_{2N}=k(\hat{\Pi},\hat{\Gamma})$。因此，$b(\hat{\Pi},\hat{\Gamma})-c(\hat{\Pi},\hat{\Gamma})+k(\hat{\Pi},\hat{\Gamma})=b_{2N}-c_{2N}+p_{2N}+r_{2N}+\left\lceil\frac{s_{2N}}{2}\right\rceil\geqslant b_0-c_0+p_0+r_0+\left\lceil\frac{s_0}{2}\right\rceil=b(\Pi,\Gamma)-c(\Pi,\Gamma)+p(\Pi,\Gamma)+r(\Pi,\Gamma)+\left\lceil\frac{s(\Pi,\Gamma)}{2}\right\rceil$。

现在证明，存在一个加帽 $\hat{\Gamma}$，使得

$$b(\hat{\Pi},\hat{\Gamma})-c(\hat{\Pi},\hat{\Gamma})+k(\hat{\Pi},\hat{\Gamma})=b(\Pi,\Gamma)-c(\Pi,\Gamma)+p(\Pi,\Gamma)+r(\Pi,\Gamma)+\left\lceil\frac{s(\Pi,\Gamma)}{2}\right\rceil$$

该加帽 $\hat{\Gamma}$ 构造 $2N$ 条连接 $G(\Pi,\Gamma)$ 中的 Π 帽子和 Γ 尾部的灰边 g_1，…，g_{2N} 的序列，且使 $\Delta_i=0 \quad 1\leqslant i\leqslant 2N$。

假定前 $i-1$ 条这样的边已经被找到，设 G_{i-1} 是向 $G(\Pi,\Gamma)$ 添加这 $i-1$ 条边的结果。如果 G_{i-1} 有一条 $\Pi\Pi$ 路，那么它也有一条 $\Gamma\Gamma$ 路，并且根据引理 10.30，存在一条染色体间的或者定向的灰边，把这些路连接成为一条定向的 $\Pi\Gamma$ 路。显而易见，对这条边有 $\Delta c=-1$，$\Delta p=-1$，$\Delta r=0$，$\Delta s=0$，这表明 $\Delta=0$。

如果 G_{i-1} 有至少有两个半结点（即 $s_{i-1}>1$），设 v_1 和 v_2 是在不同半结点上的一个 $\Pi\Gamma$ 顶点和一个 $\Gamma\Pi$ 顶点。如果 v_1 和 v_2 在 Π 的不同染色体中，那么灰边 $g_i=(v_1,v_2)$“消除”双方的半结点。因此 $\Delta c=-1$，$\Delta p=0$，$\Delta r=0$，$\Delta s=-2$，$\Delta=0$。如果 v_1 和 v_2 属于同一染色体，那么根据引理 10.31 存在一条定向灰边，把这些路连接成为一条定向的 $\Pi\Gamma$ 路。这条灰边消除两个半结点。因此，在这种情形下也有 $\Delta=0$。

如果 G_{i-1} 只有一个半结点，设 P_1 是这个半结点中的一条 $\Pi\Gamma$ 路。如果在这个半结点中仅有一条 $\Pi\Gamma$ 路，那么对“封闭”这条路的边 g_i，有 $\Delta c=0$，$\Delta p=0$，$\Delta r=1$，$\Delta s=-1$，表明 $\Delta=0$。否则，$\Delta c=0$，$\Delta p=0$，$\Delta r=0$，$\Delta s=0$，表明 $\Delta=0$。

如果 G_{i-1} 既没有一条 $\Pi\Pi$ 路也没有一个半结点，设 g_i 是一条封闭 G_{i-1} 中任意一条 $\Pi\Gamma$ 路的边。因为 g_i 不属于半结点，有 $\Delta c=0$，$\Delta p=0$，$\Delta r=0$，$\Delta s=0$，$\Delta=0$。因此，构造的边 g_1，…，g_{2N} 的序列把 $G(\Pi,\Gamma)$ 变换成 $G(\hat{\Pi},\hat{\Gamma})$，使得 $b(\hat{\Pi},\hat{\Gamma})-c(\hat{\Pi},\hat{\Gamma})+k(\hat{\Pi},\hat{\Gamma})=b(\Pi,\Gamma)-c(\Pi,\Gamma)+p(\Pi,\Gamma)+r(\Pi,\Gamma)+\left\lceil\frac{s(\Pi,\Gamma)}{2}\right\rceil$。 ■

因为 $0\leqslant f(\Pi,\Gamma)\leqslant 1$，引理 10.29 和定理 10.14 表明 $b(\Pi,\Gamma)-c(\Pi,\Gamma)+p(\Pi,\Gamma)+r(\Pi,\Gamma)+\left\lceil\frac{s(\Pi,\Gamma)}{2}\right\rceil$ 是在一个不相关的基因组的重排内，其基因组距离为 $d(\Pi,\Gamma)$。在下面的章节中，对于任意基因组，缩小 $b(\Pi,\Gamma)-c(\Pi,\Gamma)+p(\Pi,\Gamma)+r(\Pi,\Gamma)+\left\lceil\frac{s(\Pi,\Gamma)}{2}\right\rceil$ 和 $d(\Pi,\Gamma)$ 之间的差距。

10.15 基因组距离的对偶定理

缩小 $b(\Pi,\Gamma)-c(\Pi,\Gamma)+p(\Pi,\Gamma)+r(\Pi,\Gamma)+\left\lceil\frac{s(\Pi,\Gamma)}{2}\right\rceil$ 和 $d(\Pi,\Gamma)$ 之间差距的主要困难是“无覆盖（uncovering）”在对偶定理中保持“障碍（obstacle）”。原来对偶定理包括 7(!) 个参数，使得它后面隐藏的直觉知识变得非常难以解释。

定理 10.14 对于前 5 个参数提供这样的一个直觉。下面定义两个附加参数。

$G(\Pi, \Gamma)$ 中包含一条 $\Pi\Gamma$ 路的分支是简单的，如果它不是一个半结点。

引理 10.32 *存在一个最佳加帽 $\hat{\Gamma}$，使得在简单分支中封闭全部 $\Pi\Gamma$ 路。*

设 $\bar{G}$ 是通过封闭简单分支中的全部 $\Pi\Gamma$ 路而从 $G(\Pi, \Gamma)$ 得到的图。为避免混乱，在 $\bar{G}$ 中使用术语真结点（real-knot），超真结点（super-real-knot）和真结点堡垒（fortress-of-real-knot），同时定义 $rr(\Pi, \Gamma)$ 是 $\bar{G}$ 中真结点的数目。注意 $rr(\Pi, \Gamma)$ 不必与 $r(\Pi, \Gamma)$ 一致。

相关的基因组 Π 和 Γ 形成一个弱真结点堡垒（weak-fortress-of-real-knot），如果①它们在 $\bar{G}$ 中有奇数个真结点；②这些真结点中的一个是 $\bar{G}$ 中最大的真结点；③除去最大的那个以外的每个真结点是 $\bar{G}$ 中的超真结点；④$s(\Pi, \Gamma)>0$。注意到一个弱真结点堡垒通过封闭包含在这些半结点之一中的 $\Pi\Gamma$ 路能够被变换成一个真结点堡垒。

两个附加参数定义如下

$$fr(\Pi,\Gamma)=\begin{cases}1, & \text{如果 } \Pi \text{ 和 } \Gamma \text{ 在 } \bar{G} \text{ 中形成一个真结点堡垒或一个弱结点堡垒} \\ 0, & \text{其他}\end{cases}$$

```
Genomic-Sort(Π,Γ)
构建图 G=G(Π,Γ)
封闭 G=G(Π,Γ)的简单分支中的所有 ΠΓ 路(引理 10.32)
封闭包含多于一条 ΠΓ 路的分支中除一条以外的所有 ΠΓ 路
while G 包含一条路
    if G 中存在一条 ΠΠ 路
        找出一条染色体间的或一条定向的边 g,将 ΠΠ 路和 ΓΓ 路相连接(引理 10.30)
    else if G 有多于 2 个半结点
        找出一条染色体间的或定向的边 g 连接任两个半结点内的 ΠΓ 路(引理 10.31)
    else if G 有 2 个半结点
        if G 有最大真结点
            找出一条边 g,封闭这些半结点之一中的 ΠΓ 路
        else
            找出一条染色体间的或一条定向边 g 连接这些半结点中的 ΠΓ 路(引理 10.31)
    else if G 有 1 个半结点
        找出一条边,将这些半结点中 ΠΓ 路封闭
    else
        找出一条边 g,将任意 ΠΓ 路封闭
    在图 G 中增加一条边 g,即 G←G+{g}
找出一个由图 G=G(Π,Γ)定义的加帽 Γ̂
将基因组 Π̂ 排序变成 Γ̂(定理 10.14)
将 Π̂ 排序变成 Γ̂ 模拟了将 Π 排序变成 Γ
```

图 10.20 Genomic-Sort 算法

$$gr(\Pi,\Gamma)=\begin{cases}1, & \text{如果在 } \bar{G} \text{ 中存在最大的真结点并且 } s(\Pi,\Gamma)>0\\ 0, & \text{其他}\end{cases}$$

下面的定理证明 Genomoc-Sort（见图 10.20）解决基因组的排序问题。Genomic-Sort 的运行时间是 $O(n^4)$（由按反序排序有符号排列的运行时间控制），其中 n 是全部基因的数目。

定理 10.15 （Hannenhalli 和 Pevzner[153]）

$$d(\Pi,\Gamma)=b(\Pi,\Gamma)-c(\Pi,\Gamma)+p(\Pi,\Gamma)+rr(\Pi,\Gamma)+\left\lceil\frac{s(\Pi,\Gamma)-gr(\Pi,\Gamma)+fr(\Pi,\Gamma)}{2}\right\rceil。$$

10.16 基因组重复

整个基因组加倍是复制过程中经常出现的致命事故。然而，如果这种加倍可以在生物体中被分解并且在一个个体中固定为正常状态，那它表现为全基因组的重复。这样一种事件甚至可导致进化优势，因为加倍的基因组拥有每个基因的 2 个拷贝，它们可以独立地进化。因为基因可能发展异常功能，基因组重复（genome duplication）可能引起迅速的进化进程。例如，两亿年前脊椎动物基因组经历过重复（Ohno 等[256]），在一些脊椎动物谱系上出现更多新的重复（Postlethwait[278]）。植物基因组的比较遗传图也显示多重重复（Paterson 等[260]）。

酵母测序揭示了关于酵母基因组一亿年前发生的一次远古加倍的证据（Wolfe 和 Shields[369]）。最初，复制出的基因组包含每个染色体的两个同样拷贝，但经过倒位、易位、融合和裂变，这 2 个拷贝遭到破坏。El-Mabrouk 等[96]，提出了古酵母基因组加倍前基因次序重构问题的解决方案。

一个重排的复制基因组包含每个基因的 2 个拷贝。基因组重复问题是为了计算把一个重排的复制基因组变换成一些理想的复制基因组（perfect duplicated genome）所需的易位的最少次数，这些理想的复制基因组具有偶数条染色体，包含了每条染色体的 2 个同样拷贝。例如，由 4 条染色体组成的重排复制基因组 $\{abc, def, aef, dbc\}$ 通过染色体 aef 和 dbc 之间的一次易位就能够变换成一个理想的复制基因组 $\{abc, def, abc, def\}$。El-Mabrouk 等[96]提出了一个多项式算法，解决重排操作只有易位情形的基因组重复问题。这个算法利用了 Hannenhalli[151]关于多染色体基因组之间易位距离的对偶定理。如何设计更适当的既有易位又有反序的基因组重复分析的问题仍未得到解决。

10.17 若干其他的问题和方法

10.17.1 基因组重排和系统发育研究

Sankoff 等[304]第一次将重排距离运用到分子进化的研究中。基因组距离问

题在多基因组方面的推广相应于下列各项。

多基因组的距离问题 给出一组排列 π^1，…，π^k，寻找一个排列 σ，使得 $\sum_{i=1,k} d(\pi^i,\sigma)$ 最小（d 是基因组 π^i 和 σ 之间的距离）。

在 $d(\pi,\sigma)$ 是 π 和 σ 之间的反序距离的情形下，多基因组距离问题已被证明是 NP-难题（Caprara[56]）。类似于进化树多重联配，存在预先不知道系统树的情形下的多基因组距离问题的一个推广（见图 10.21）。因为多基因组距离在反序距离的情形下是困难的，所以多数基因组分子进化研究基于断点距离。排列 π 和 σ 之间的断点距离 $d(\pi,\sigma)$ 定义为 $\pi\sigma^{-1}$ 上的断点数目。虽然这种陈述中的多基因组距离也是 NP-难题，但 Sankoff 和 Blanchette[301] 对这个问题提出了一个有实用价值的启法式算法。

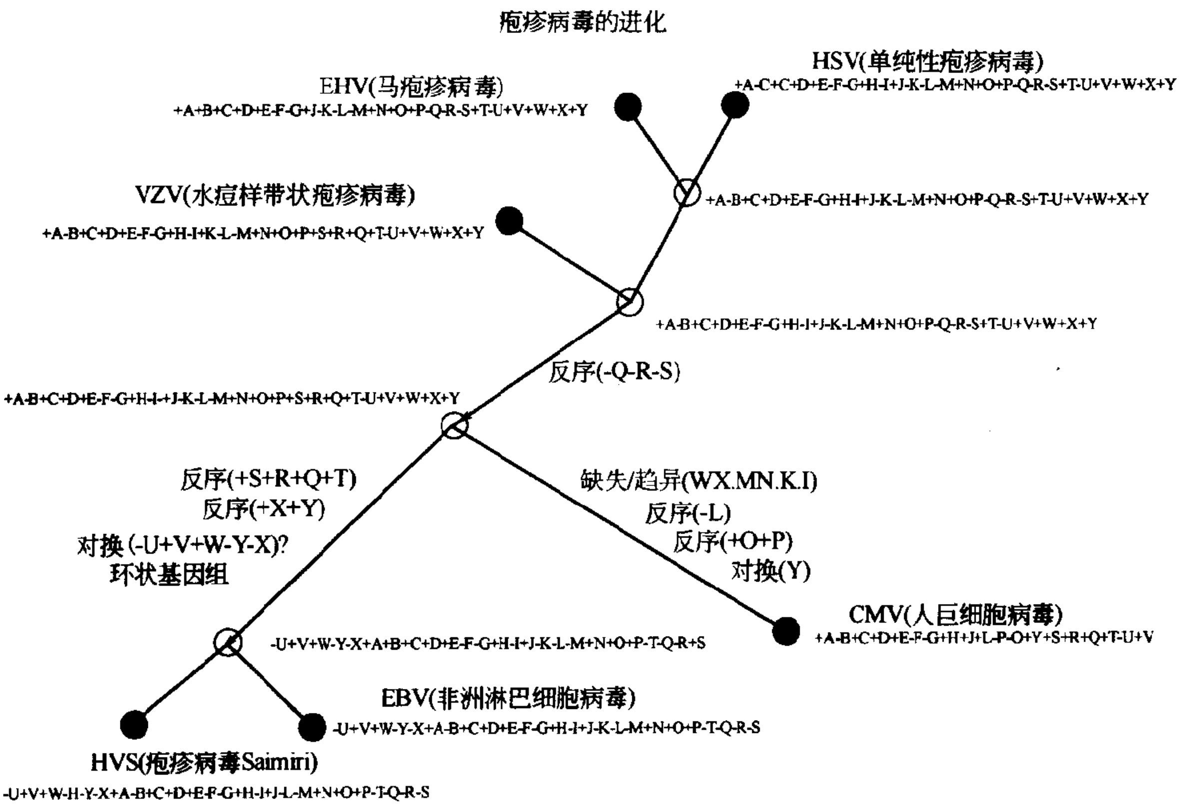


图 10.21 疱疹病毒进化的推断方案

10.17.2 依照反序排序的快速算法

Berman 和 Hannenhalli[33] 和 Kaplan 等[185] 对于按反序排序有符号的排列设计了一个快速算法。Kaplan 等[185] 二次算法跳过 Hannenhalli-Pevzner 算法中的等价变换步骤，并且探索了灰边交叉图的特性（而不是圈的交叉图）。这可以导出定理 10.10 的一个更优雅、简单的证明。

11 计算蛋白质组学

11.1 引言

质谱仪能在几秒钟内将肽打碎成若干片段，并且测出它们的质量（肽谱）。肽测序问题是指：给定肽谱，推导出肽的序列。对于一个理想的断裂过程（每一个肽都是在两个相连的氨基酸之间断开）和一个理想的质谱仪来说，肽测序问题是很简单的。但实际上，断裂的过程远没有那么理想，这给肽的从头测序（*de novo* peptide sequencing）带来困难。

数据库搜索是解决肽从头测序问题的一种可供选择的方法，而且质谱方法也已经能够成功地识别出基因组数据库中已有的蛋白质[261]。在研究众多已测序生物体的基因组时，用质谱进行数据库搜索[97,101,230,335]主要依赖于“在书后寻找答案”的能力。将一个实验谱与数据库中每一个肽的理论谱作比较，数据库中最符合实验谱的肽的序列通常就是实验所要测的肽的序列。特别是 Eng 等人[97]在 1994 年从Ⅱ类 MHC（主要组织相容性复合体）复合物中识别出蛋白质，以及 Clauser 等[72]在 1999 年识别出与预惊厥效应相关的蛋白质。然而，考虑到导入质谱仪的样品具有动态性，以及可能存在的多突变和修饰，依靠精确匹配和几乎精确匹配的数据库搜索方法受到了质疑。试图解释串联质谱的从头预测算法，在缺少数据库时，是识别未知蛋白质的重要方法。但是，这个方法在具有高质量谱的时候是最有效的。

由于蛋白质是细胞信号传导和新陈代谢调节这个复杂系统的一部分，所以它们也会受到不计其数的生物修饰（诸如磷酸化和糖基化）以及遗传变异[134,207]等的影响。例如，人类基因组中至少存在 1000 种激酶，这表明磷酸化对于信号传导和酶的活化是普遍的机制。几乎所有的蛋白质序列都是翻译后修饰的，且有 200 种氨基酸残基的修饰是已知的。由于现在还不能从 DNA 序列预测出翻译后的修饰，所以这在人类基因组（测序）完成以后仍是公认的难题。这也就提出了后基因组时代的一个计算难题：给定大量代表人类蛋白质组的谱，查明每个人类基因中出现的修饰是 200 种修饰中的哪一种。

自从 Biemann 和 Scoble[35]在 1987 年发表了那篇经典的论文以后，通过质谱技术识别修饰蛋白质已有若干成功的事例。Mann 和 Wilm[230]及 Yates 等[373,374]首先对修饰肽进行了计算分析。由于质谱技术有时将化学修饰引入了天然肽，使得肽对于数据库搜索程序变得“不可见”，所以对修饰肽进行计算分析非常重要。

Mann 和 Wilm[230]提出的肽序列标签法（peptide sequence tag approach）中将部分从头预测算法和数据库搜索结合起来。一个肽序列标签是肽中一段短的、能明显识别的子字符串，用它可减少对包含该标签的肽的搜索。Yates 等[374]提出一个穷举搜索方法，它针对小部分修饰（隐含地）产生一个所有修饰肽的虚拟数据库，并将谱和这个虚拟数据库相匹配。这产生了一个很大的组合问题，甚至是对小部分修饰类型也是如此。另外的一个局限是一些极其巨大的改变，如糖基化，会打乱碎片的（特征序列）模式，这时就不能用这种方法来分析了。

质谱技术中的容许突变数据库（mutation-tolerant database）搜索可简述如下：给定一实验谱，要在一些肽中找出和该谱最匹配的肽，这些肽除了数据库中的肽，还可以包括至多 k 个突变。这并不是个简单的问题，因为非常相似的肽可以有完全不同的谱。Pevzner 等[270]提出了谱相似性的概念并由此导出一个能识别相关频谱的算法，即使对应的肽有多重修饰或突变。这种算法不通过穷举搜索找出潜在的肽修饰，所以不需要产生一个修饰肽的虚拟数据库。

尽管数据库搜索非常有用，但是生物学家想要在质谱数据的基础上克隆一条新基因，需要的是从头预测方法，而不是数据库匹配算法。然而直到现在，通过质谱技术进行测序还没有大范围的实行，这对发现新蛋白质的影响有限。单单依靠质谱得到的序列信息来克隆基因的例子几乎没有[223]。

与质谱数据的自动采集相结合，从头肽测序的最新进展可能为“蛋白质组测序”打开一扇大门。将蛋白质与具有不同特异性的蛋白水解酶混合，经处理得到一组相互重叠的肽，然后进行测序并装配，就可以得到长的蛋白质序列。通过在单个蛋白质上应用这种方法已经证明了它能确定完整的蛋白质序列[166]。

11.2 肽测序问题

设 A 为氨基酸集合，氨基酸的分子量记为 $m(a)$，$a\in A$。肽 $P=p_1$，…，p_n 表示一个氨基酸序列，其（母体）分子质量为 $m(P)=\sum m(p_i)$。肽段（partial peptide）P'是指 P 的子序列 p_i，…，p_j，其质量为 $\sum\limits_{i\leqslant t\leqslant j} m(P_t)$。

串联质谱仪（tandem mass-spectrometer）中的肽碎片可以用一组代表离子类型（ion-type）的数 $\Delta=\{\delta_1, \cdots, \delta_k\}$ 来表示。肽段 $P'\subset P$ 的一个 δ-离子是 P'的一个修饰，质量为 $m(P')-\delta$。对于串联质谱仪来说，从 P 的所有肽段质量（在理论谱中每一个肽段产生 k 质量）中减去所有可能的离子类型 $\delta_1,\cdots,\delta_k$，可以算出肽 P 的理论谱。一个（实验）谱 $S=\{s_1,\cdots,s_m\}$ 是一组（碎片）离子质量。谱 S 与肽 P 之间的匹配是用实验谱与理论谱之间的共有质量的个数（共有波峰的个数）来衡量的。Dancik 等[79]提出了下面的问题。

肽测序问题 给定谱 S，离子类型集合 Δ，以及质量 m，找出一个质量为 m

的肽，最大程度地匹配谱 S。

记 N 末端肽段 p_1，…，p_i 为 P_i，记 C 末端肽段 p_{i+1}，…，p_n 为 P_i^-，$i=1$，…，n。实际上，串联质谱技术（MS/MS）获得的谱主要由一些 N 末端肽段和 C 末端肽段的 δ-离子所组成。为了反映这一点，理论 MS/MS 谱只由 N 末端和 C 末端肽段的离子组成（见图 11.1）。例如，最常见的 N 末端离子通常为 b 离子（b_i 对应于 $\delta=-1$ 时的 P_i），最常见的 C 末端离子通常为 y 离子（y_i 对应于 $\delta=19$ 时的 P_i^-）。离子阱质谱仪中其他常见的 N 末端离子（a，b-H_2O 和 b-NH_3）

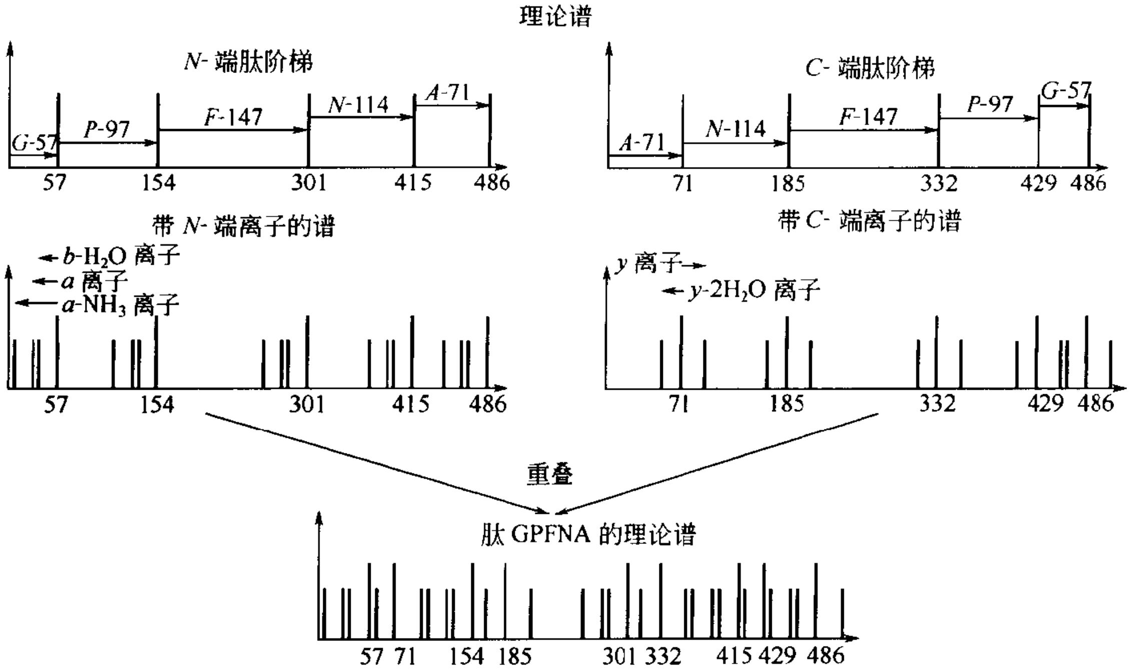


图 11.1 肽 GPFNA 的理论 MS/MS 谱

其母体分子质量为 57+97+147+114+71=486

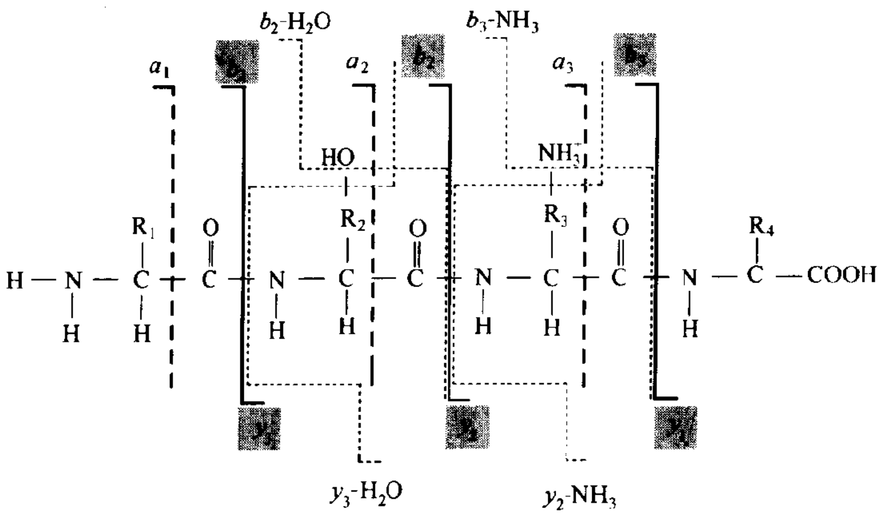


图 11.2 串联质谱中的典型断裂模式

如图 11.2 所示。此外，现有的数据库搜索和从头算法使用了更复杂的目标函数（例如带有权重的共有峰计数）来代替共有峰计数。

11.3 谱图

肽测序算法沿着穷举搜索或谱图范例两条思路发展，前一个方法（Sakurai 等，1984[294]）要产生所有氨基酸序列和相应的理论谱。其目的是找出一条理论谱与实验谱匹配最好的序列。由于可能的序列个数随着肽的长度呈指数型增长，各种不同的剪除技术被设计用来限制全局方法中的组合爆炸。前缀剪切掉[149,182,375,378]对那些前缀和实验谱匹配很好的序列的计算空间作了限制。前缀法的难点在于剪切往往丢失了一些前缀在谱中表达很低的正确序列。另外一个问题是谱信息只有在可能的肽序列产生后才能使用。

谱图方法的效果更好，因为该方法在任何候选序列被评估之前就利用了谱信息。在这种方法中，谱中的峰被转化为一种谱图[26,79,103,335]。谱中的峰作为谱图中的顶点，而谱图中的边对应了顶点间的连接，这些顶点的氨基酸质量是不同的。在实验谱中的每一个峰都被转化为谱图中的若干个顶点；每个顶点代表了峰的碎片离子类型的一个可能的分布。肽测序问题就被归结为在由谱得到的有向无圈图中找寻一条最长路。由于在有向无圈图中找寻一条最长路的有效的算法是已知的[75]，这种方法可以有效地将所有肽的集合删裁为谱图中高得分路径的集合。

图 11.3 举例说明了谱图方法。由于谱中“有意义的”峰是由肽阶梯（肽段的质量）通过 δ-移位得到的，人们可能会想到由逆 δ-移位重构理想谱，从而导出肽序列。图 11.3 说明这种做法并不可行，需要更仔细地（基于谱图概念）分析逆 δ-移位。

为了简单起见，假设一个 MS/MS 谱 $S=\{s_1,\cdots,s_m\}$ 主要由 N 末端离子组成，将其转化成谱图 $G_\Delta(S)$[26]。谱图的顶点是表示肽段可能质量的整数 $s_i+\delta_j$。谱图中的每一个峰 $s\in S$ 产生 k 个顶点 $V(s)=\{s+\delta_1,\cdots,s+\delta_k\}$。一个谱图的顶点集合是 $\{s_{\text{initial}}\}\cup V(s_1)\cup\cdots\cup V(s_m)\cup\{s_{\text{final}}\}$，其中 $s_{\text{initial}}=0$，$s_{\text{final}}=m(p)$。两个顶点 u 和 v 由一条有向边连接，如果 $v-u$ 是某氨基酸的质量，则边的方向为 u 指向 v，且边被标记为该氨基酸。如果把顶点看作是可能的 N 末端肽段，那么 u 到 v 的边表明，在 v 处的序列可以通过将 u 处的序列延伸一个氨基酸而得到（见图 11.4 和图 11.5）。

一个肽 P 的谱 S 称为是完全的，如果对于每一个 P_i，$1\leqslant i\leqslant n$，S 至少包含一个对应的离子类型。谱图的应用是基于这样一个事实，即对于一个完全谱，在 $G_\Delta(S)$ 中存在一条长度为 n，从 s_{initial} 到 s_{final} 的路径，并被 P 所标记。这个事实使得串联质谱肽测序问题变成一个在所有路径集合中寻找一条正确路径的问题。

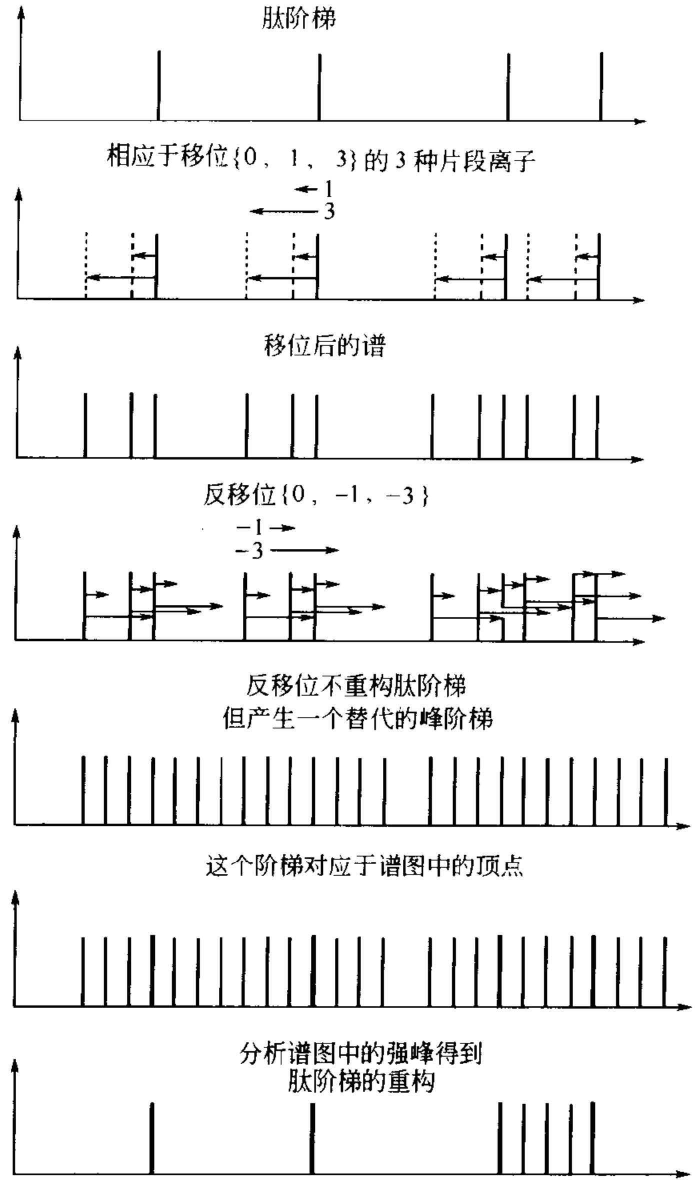


图 11.3 逆移位产生一组谱图中的顶点

“强峰”（用实线段表示）对应于经过多次逆移位得到的峰

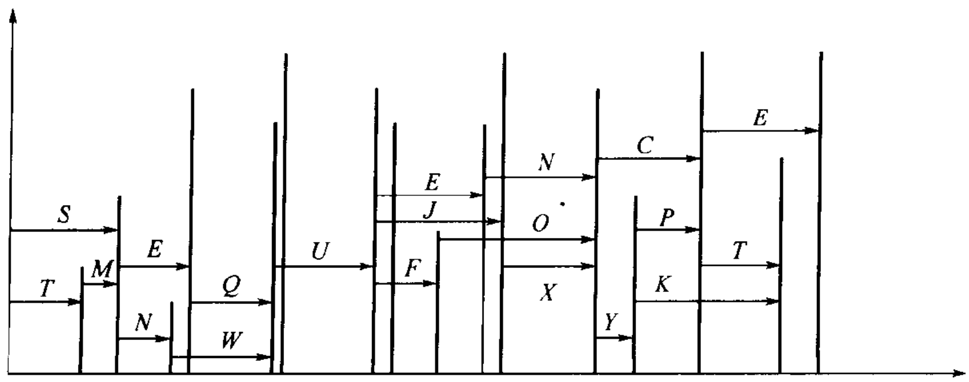


图 11.4 谱图中的多重路径

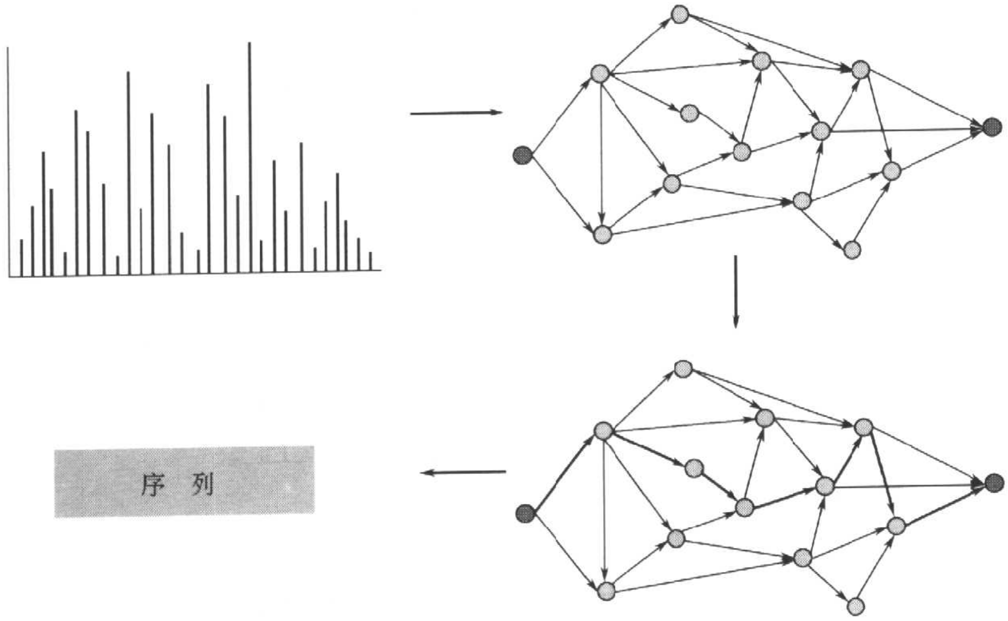


图 11.5　谱中的噪声在谱图中产生许多“假”边、顶点及对应于真肽序列的伪边。序列重构对应于在谱图中寻找最优路径

不幸的是，实验谱往往是不完全的。另一个问题是，同一肽在不同的质谱仪中做 MS/MS 实验所产生的谱差别相当大。不同的离子化方法对于产生特定的碎片离子类型的倾向性有显著的影响。因此，每一种肽测序算法都要根据特定的质谱仪进行调整。为解决这个问题，Dancik 等[79]描述了一种算法，能从已知序列的实验谱样品中自动地学习离子类型。他们引入偏频函数（offset frequency function）来估计离子类型对特定质谱仪的倾向性，推导出一种独立于质谱仪的肽测序算法。

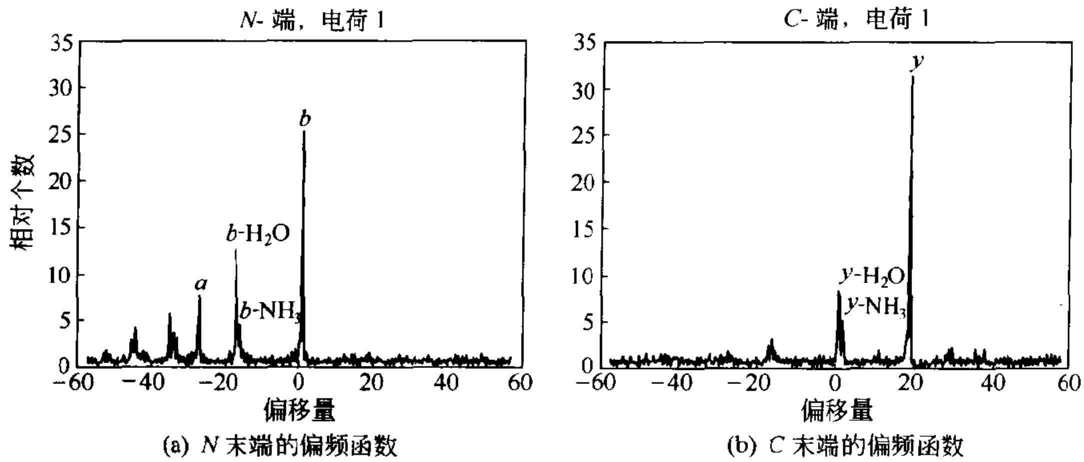
11.4　学习离子类型

如果一个给定质谱仪所产生的离子类型 $\Delta=\{\delta_1, \cdots, \delta_k\}$ 是未知的，谱图就不能被解释。下面说明在对碎片模式没有任何先验知识的情况下，如何从一个已知序列的实验谱学习得到集合 Δ 和离子倾向性。

令 $S=\{s_1, \cdots, s_m\}$ 是对应于肽 P 的一个谱。一个肽段 P_i 和一个峰 s_j 有一个偏移量 $x_{ij}=m(P_i)-s_j$；将 x_{ij} 视为一个随机变量。由于对应于“真”碎片的偏移量的概率远大于随机偏移的概率，所以在偏移量的经验分布中的峰揭示了碎片离子。对所有离子和所有肽段的偏移量进行的统计为离子类型提供了一个可靠的学习算法[79]。

给定谱 S，偏移量 x 和精度 ε，令 $H(x, S)$ 是数对(P_i, s_j) 的个数，$i=1, \cdots, n-1$，$j=1, \cdots, m$，(P_i, s_j) 对于 x 在 ε 范围内有偏移量 $m(P_i)-s_j$，

偏频函数定义为 $H(x)=\sum_S H(x, S)$，这里的求和包括了学习样本中的所有谱（见图 11.6）。为学习 C 末端离子，我们可以同样地处理数对 (P_i^-, s_j)。对应于 $H(x)$ 中的峰，偏移量 $\Delta=\{\delta_1, \cdots, \delta_k\}$ 表示由给定的质谱仪所产生的离子类型。



(a) N 末端的偏频函数　　(b) C 末端的偏频函数

图 11.6　N 末端肽段的偏频函数和 C 末端的偏频函数
横轴表示谱中峰的偏移量和肽段的分子量。竖轴表示标准化的偏移量个数，平均个数为 1

峰在谱中以强度来区分，所以在将谱转化为谱图之前必须设置一个阈值，以便在谱的噪声中辨别信号。低阈值会使谱图过分增长，而高阈值则会使谱图断裂。偏频函数允许人们严格地设置强度阈值[79]。

11.5　给谱图中的路径打分

打分的目的是为了量化候选肽"解释"谱的能力，并选出能最好地解释谱的肽。如果 $p(P, S)$ 是肽 P 产生谱 S 的概率，那么打分的目的就是找出一个肽 P，对于给定的谱 S，使得 $p(P, S)$ 最大。下面描述一个概率模型来估计 $p(P, S)$，并且导出一个给谱图中的路径打分的方案。在加权的谱图中，最长路径就对应着能够最好地解释谱 S 的肽 P。

在概率方法中，串联质谱用一组离子类型 $\Delta=\{\delta_1, \cdots, \delta_k\}$ 和它们的概率 $\{p(\delta_1), \cdots, p(\delta_k)\}$ 来刻画，肽段的 δ_i 离子以概率 $p(\delta_i)$ 独立地产生。质谱仪同时会产生"随机的"（化学或电子）噪声，以概率 q_R 在任何位置产生峰。因此，在某一位置以概率 $q_i=p(\delta_i)+(1-p(\delta_i))q_R$ 产生对应于某一 δ_i 离子的峰，这个概率可以由观测经验分布估计出来。在理论上，一个肽段在谱中可以有至多 k 个相应的峰。产生所有 k 个峰的概率为 $\prod_{i=1}^{k} q_i$，没有峰产生的概率是 $\prod_{i=1}^{k}(1-q_i)$。概率模型定义一个肽 P 产生一个谱 S 的概率为 $p(P, S)$。下面描述如何计算 $p(P, S)$，并且导出打分方案，从而对于给定的谱 S，找到一条肽使得 $p(P, S)$

最大化。

假设一个候选肽段 P_i 在谱 S 中产生离子 δ_1，…，δ_l（“呈现”离子），不产生离子 δ_{l+1}，…，δ_k（“丢失”离子）。这 l 个“呈现”离子将会在与 P_i 对应的谱图中产生一个顶点，如何给这个顶点打分呢？一个直观的方法是使用“被解释离子加分”的方法，即这个顶点的分数和 q_1，…，q_l 成比例，或者和 $\frac{q_1}{q_R}$，…，$\frac{q_l}{q_R}$ 成比例，后者使得概率除去噪声而标准化。然而，这种方法有一个严重的缺陷，若采用“呈现离子加分，丢失离子罚分”方法，结果就会显著改进。顶点的（概率）得分给出如下

$$\frac{q_1}{q_R}\cdots\frac{q_l}{q_R}\frac{(1-q_{l+1})}{(1-q_R)}\cdots\frac{(1-q_k)}{(1-q_R)}$$

我们认为这个公式的作用是解决了二肽 GG 和具有相同质量的氨基酸 N 之间的简单替换。在对丢失离子不罚分的情况下，GG 被选来代替 N 表示任何支撑第一个 G 的位置上的峰（即使是对应于随机噪声的一个极弱的峰）。这种规则会产生许多错误的 GG-丰富（GG-abundant）的解释，由此表明一个好的规则会选出被具有足够强度的主要离子类型所支撑的 GG，而这是由“呈现离子加分，丢失离子罚分”的打分方法自动地执行的。

为了简单起见，假设所有的肽段都具有相同的可能性，且忽略峰的强度。把所有质量的空间离散成从 0 到母体质量 $m(P)=M$ 的区间，记 $T=\{0, \cdots, M\}$，并且将谱表示为一个 M 维向量 $S=\{s_1, \cdots, s_M\}$，其中 s_t 表示位置 t 是否存在峰（若位置 t 存在的峰则 $s_t=1$，否则 $s_t=0$）。对于一个给定的肽 P 和位置 t，s_t 是一个 0-1 随机变量，其概率分布为 $p(P, s_t)$。

令 $T_i=\{t_{i1}, \cdots, t_{ik}\}$ 是表示肽段 P_i 的 Δ 离子的位置集合，其中 $\Delta=\{\delta_1, \cdots, \delta_k\}$。令 $R=T\setminus\bigcup_i T_i$ 是与任何肽段都没有联系的位置集合。概率分布 $p(P, s_t)$则视 $t\in T_i$ 还是 $t\in R$ 而定。若位置 $t=t_{ij}\in T_i$，则概率 $p(P, s_t)$ 为

$$p(P, s_t)=\begin{cases}q_j, & \text{如果 } s_t=1\text{（在 } t \text{ 位置有峰）}\\ 1-q_j, & \text{其他情况}\end{cases}$$

类似地，若 $t\in R$，则概率 $p(P, s_t)$ 为

$$p_R(P, s_t)=\begin{cases}q_R, & \text{如果 } s_t=1\text{（在 } t \text{ 位置有随机噪声）}\\ 1-q_R, & \text{其他情况}\end{cases}$$

谱中全部“噪声”峰的概率为 $\prod_{t\in R}p_R(P, s_t)$。

令 $p(P_i, S)=\prod_{t\in T_i}p(P, s_t)$ 是肽 P_i 在集合 T_i（所有其他的位置都被忽略）中的位置上产生给定谱的概率。为了简单起见，假设谱中的每个峰只属于一个集合 T_i，并且所有的位置都是独立的。那么

$$p(P,S)=\prod_{t=1}^{M}p(P,s_t)=(\prod_{i=1}^{n}p(P_i,S))\prod_{t\in R}p_R(P,s_t)$$

对于给定的谱 S，值 $\prod_{t\in T}p_R(P,s_t)$ 与 P 无关，并且 $p(P,S)$ 的最大化等价于

$$\frac{p(P,S)}{p_R(S)}=\frac{\prod_{i=1}^{n}\prod_{j=1}^{k}p(P,s_{t_{ij}})\prod_{t\in R}p_R(P,s_t)}{\prod_{t\in T}p_R(P,s_t)}=\prod_{i=1}^{n}\prod_{j=1}^{k}\frac{p(P,s_{t_{ij}})}{p_R(P,s_{t_{ij}})}$$

的最大化，其中 $p_R(S)=\prod_{t\in T}p_R(P,s_t)$。

在对数尺度里，上面的公式意味着对谱图中的顶点做附加的“呈现离子加分，丢失离子罚分”打分[79]。

11.6 肽测序与反对称路径

在构造加权谱图以后，将肽测序问题归结为有向无圈图中的最长路径问题，这个问题可由快速线性时间算法解决。不幸的是，这种简单的算法在实践中不太有用。问题在于谱中每一个峰可能被解释为 N 末端离子或者 C 末端离子。因此每一个“真”顶点（对应于质量 m）都有一个“假”双顶点［对应于质量 $m(P)-m$］。此外，如果真顶点有很高的得分，那么它的假双顶点的得分也很高。于是谱图中的最长路径会包括真顶点和假双顶点，因为它们都有高的得分。这种路径并不对应于切实可行的蛋白质重构，所以必须避免。然而已知的最长路径算法并不能避免这种路径。

因此，将肽测序问题简化为前面提到的最长路径问题是不合适的。下面简述反对称最长路径问题，这是解决肽序列重构问题的恰当模型。

设 G 为图，且设 T 为 G 的（双）顶点的一组禁用对（forbidden pairs）。G 中的路径称为反对称的，如果它至多包含每个禁用对中的一个顶点。反对称最长路径问题是在带一组禁用对 T 的图 G 中找出最长的反对称路径。不幸的是，反对称最长路径问题是 NP 难题[119]，这表明解决这个问题的有效算法是不可能有的。不过，这个负面结果并不表示没有希望找到解决串联质谱肽测序的有效方法，因为这个问题有禁用对的特殊结构。

谱图中的顶点按数字来构造，这些数字对应于可能的肽段的质量。如果区间 (x_1, y_1) 和 (x_2, y_2) 不交错，顶点的两个禁用对 (x_1, y_1) 和 (x_2, y_2) 称作是非交错的（non-interleaving）。如果顶点的每个禁用对都是非交错的、带有一组禁用对的图 G 称为是真的（proper）。

串联质谱肽测序问题对应于真图中的反对称最长路径问题。Dancik 等[79]证明了存在解决真图中反对称最长路径问题的有效算法。

11.7 肽鉴定问题

Pevzner 等[270]研究了下面这个问题。

肽鉴定问题 给定一个肽数据库、谱 S、离子类型集合 Δ 和参数 k，找寻一个肽，使其最大程度地匹配谱 S，并且与原数据库中的条目相比有不超过 k 处的突变和修饰。

肽鉴定问题的主要困难是非常相似的两个肽 P_1 和 P_2 可能会有两个完全不同的谱 S_1 和 S_2。我们的目标是给出一个和序列相似性关联的谱相似度的定义。换句话说，如果 P_1 和 P_2 分别只有若干处替换、插入、删除或修饰的话，谱 S_1 和 S_2 的相似度就会很高。显然，从直觉上可以想到用共有峰的个数来度量谱相似度。然而这个度量随着突变数量的增加而迅速下降，从而导致在 MS/MS 数据库搜索中检测相似度时带有局限性。此外，相关肽的谱之间还有很多关联性，其中只有一小部分可以被“共有峰”计数方法所捕获。PEDANTA 算法[270]对任何 k 都能捕获相关谱之间所有的关联性，并可以处理肽中存在较大突变，以至改变碎片模式的情况。例如，替换某些氨基酸如 H，K，R 和 P 等可能会显著地改变断裂方式。甚至在极端的情况下——当单个突变把“只有 b 离子的”碎片模式改变成“只有 y 离子的”碎片模式——PEDANTA 算法也能够揭示出两个相关谱之间的相似性。

11.8 谱的卷积

设 S_1 和 S_2 是两个谱，定义谱的卷积 $S_2 \ominus S_1 = \{s_2 - s_1 : s_1 \in S_1, s_2 \in S_2\}$，并且令$(S_2 \ominus S_1)(x)$为集合中元素 x 的重数。换言之，$(S_2 \ominus S_1)(x)$是使 $s_2 - s_1 = x$ 成立的 $s_1 \in S_1$，$s_2 \in S_2$ 的个数。如果具有谱 S 的肽 P 的母体质量是 $M(P)$，那么 $S^R = M(P) - S$ 称为谱 S 的反谱（reversed spectrum）[S 中的每一个 b 离子（y 离子）相应于 S^R 中的 y 离子（b 离子）]。反谱的卷积（$S_2 \ominus S_1^R$）（x）是使 $s_2 + s_1 - M(P) = x$ 成立的 $s_1 \in S_1$，$s_2 \in S_2$ 的数量。

为了说明该方法的思想，考虑同一个肽的两个复制肽 P_1 和 P_2。S_1 和 S_2 之间的公共峰个数（共有峰计数）就是 $x=0$ 处 $S_2 \ominus S_1$ 的值。许多 MS/MS 数据库搜索算法都是想找到数据库中的一个肽，使得 $x=0$ 处 $S_2 \ominus S_1$ 的值最大，其中 S_2 是实验谱，S_1 是理论谱。然而，如果在 P_2 中引入了 k 个突变以区别于 P_1，则在 $x=0$ 处 $S_2 \ominus S_1$ 的值迅速下降。事实上，共有峰计数方法的鉴别能力会在 $k=1$时迅速下降，在 $k>1$ 时几乎消失。

谱卷积中的峰可以使人们不做穷举搜索就能检测到突变和修饰。设 P_2 与 P_1 仅相差一个突变（$k=1$），氨基酸质量的差异为 $\delta = M(P_2) - M(P_1)$。这时，$S_2 \ominus S_1$ 在 $x=0$ 和 $x=\delta$ 处预期会有两个近似相等的峰。如果突变在肽的 t 处发生，

那么 $x=0$ 处的峰对应于 $i<t$ 的 b_i 离子和 $i\geqslant t$ 的 y_i 离子。$x=\delta$ 处的峰对应于 $i\geqslant t$的 b_i 离子和 $i<t$ 的 y_i 离子。P_2 中的一个突变使 $M(P_1)$减少 δ，也会使谱 S_2 中的一些峰平移 δ 个单位从而使 S_2 发生“突变”。因此，S_1 与“突变的” S_2 之间的共有峰个数可能比 S_1 与 S_2 之间的共有峰个数多。增加的个数受$(S_2\ominus S_1)(\delta)$限制，且 S_1 与“突变的” S_2 之间的共有峰个数的上界是$(S_2\ominus S_1)(0)+(S_2\ominus S_1)(\delta)$。

突变肽的谱之间的另一个相关性集合可以被反谱卷积 $S_2\ominus S_1^R$ 所捕获，反映了 N 末端离子和 C 末端离子的成对性。$S_2\ominus S_1^R$ 在相同位置 0 和 δ 上预期会有两个峰。

现在假设 P_1 和 P_2 是两个不同的替换肽，其中一个的质量相差 δ_1，另一个质量相差 $\delta-\delta_1$。这些突变在谱卷积的 $x=\delta_1$ 和 $x=\delta-\delta_1$ 处产生了两个新的峰。对于随机肽中均匀分布的突变而言，在 0，δ，δ_1，$\delta-\delta_1$ 处的峰的期望高度之比为 2∶2∶1∶1。

为了增加信噪比，将谱中的峰与反谱卷积中的峰相叠加

$$S=S_2\ominus S_1+S_2\ominus S_1^R$$

并且，通过引入移位函数来叠加在 0 和 δ 处（在 δ_1 和 $\delta-\delta_1$ 处同样）的峰，移位函数为

$$F(x)=\frac{1}{2}[S(x)+S(\delta-x)]$$

注意到 $F(x)$是沿轴 $x=\frac{\delta}{2}$对称的，且有 $F(0)=F(\delta)$和 $F(\delta_1)=F(\delta-\delta_1)$。而感兴趣的是 $x\geqslant\frac{\delta}{2}$时 $F(x)$的峰。

定义 $x_1=\delta=M(P_2)-M(P_1)$和 $y=F(\delta)=F(0)$。令 $y_2=F(x_2)$，$y_3=F(x_3)$，…，$y_k=F(x_k)$是 $x\geqslant\delta/2$ 和 $x\neq\delta$ 时 $F(x)$ 的第 $k-1$ 个最大的峰。定义

$$SIM_k(S_1,S_2)=\sum_{i=1}^{k}y_i$$

为假设对应的肽之间有 k 个突变时谱 S_1 和 S_2 之间的相似性估计。SIM_k 通常是移位函数的 k 个最高峰的总体高度。例如，$SIM_1(S_1,S_2)=y_1$ 表示 P_2 中有 $k=1$个突变时 S_1 和“突变的” S_2 之间的共有峰个数的上界。

虽然谱的卷积能够帮助鉴定突变的肽，但是它还有严重的局限性，现叙述如下。

设

$$S=\{10,20,30,40,50,60,70,80,90,100\}$$

是肽 P 的谱，为简单计，假设 P 只产生 b 离子。令

$$S'=\{10,20,30,40,50,55,65,75,85,95\}$$

和

$$S''=\{10,15,30,35,50,55,70,75,90,95\}$$

为对应于数据库中肽 P'和 P''的两个理论谱。哪一条肽（P'或 P''）最匹配谱 S? 共有峰计数方法不能解决这个问题，因为 S'和 S''同 S 都有五个共有峰。而且，谱卷积也不能解决这个问题，因为 $S\ominus S'$和 $S\ominus S''$（以及相应的移位函数）会在 0 和 5 处显现出相同高度的峰。这表明 P'和 P''都可以通过单突变 P，将其质量改变 5 而得到。然而更细致的分析表明，尽管对于 P'这个单突变可以通过在质量 50 后引入位移 5 而实现，但是对于 P''这个单突变却不能实现。S'和 S''的主要区别在于 S'中的匹配位置会组成块，而在 S''中的匹配位置就不能。下面介绍谱联配方法，它可以解决这个问题。

11.9　谱联配

设 $A=\{a_1, \cdots, a_n\}$是自然数 $a_1<a_2<\cdots<a_n$ 的一个有序集。移位 Δ_i 则将 A 变为$\{a_1, \cdots, a_{i-1}, a_i+\Delta_i, \cdots, a_n+\Delta_i\}$。仅考虑那些不改变元素顺序的移位，即移位 $\Delta_i\geqslant a_{i-1}-a_i$。给定集合 $A=\{a_1, \cdots, a_n\}$ 和 $B=\{b_1, \cdots, b_m\}$，要找到一系列 A 的 k 位移，使得 A 尽可能地与 B 相似。集合 A 和 B 之间的 k-相似性（k-similarity）$D(k)$定义为经过 k 步位移后，两个集合具有相同元素的最大数。例如，移位 -5_6 将

$$S=\{10,20,30,40,50,60,70,80,90,100\}$$

变为

$$S'=\{10,20,30,40,50,55,65,75,85,95\}$$

所以 $D(1)=10$。集合 $S''=\{10,15,30,35,50,55,70,75,90,95\}$共有 5 个元素与 S 相同（S'也一样）。但是没有一个移位可以将 S 转换成 S''，并且 $D(1)=6$。下面描述一个计算 $D(k)$的动态规划算法。

定义谱积 $A\otimes B$ 是一个 $a_n\times b_m$ 的二维矩阵，矩阵中有 nm 个 1，对应于所有成对索引（a_i, b_j），其余的元素都为零。在矩阵主对角线上的 1 的个数表示谱 A 和 B 的共有峰计数，换言之，就是谱 A 和 B 的 0-相似性。图 11.7 表示了前节例子中 S, S', S''的谱积 $S\otimes S'$和 $S\otimes S''$。在图 11.7 中的两种情形中，主对角线上 1 的个数都是一样的，$D(0)=5$。平移了 δ 的峰的个数是对角线（i, $i+\delta$）上 1 的个数。移位函数的局限性是它仅考虑对角线，没有将对角线和可行突变的情况结合起来考虑。对于利用至多 $k+1$ 条对角线的谱矩阵，谱的 k-相似性定义为通过谱矩阵的路径上 1 的最大个数，k-最优谱联配（k-optimal spectral alignment）定义为利用这 $k+1$ 条对角线的路径。例如，对利用至多 2 条对角线的矩阵，1-相似性定义为通过矩阵的路径上 1 的最大个数。从图 11.7 可以看出，利用 1-相

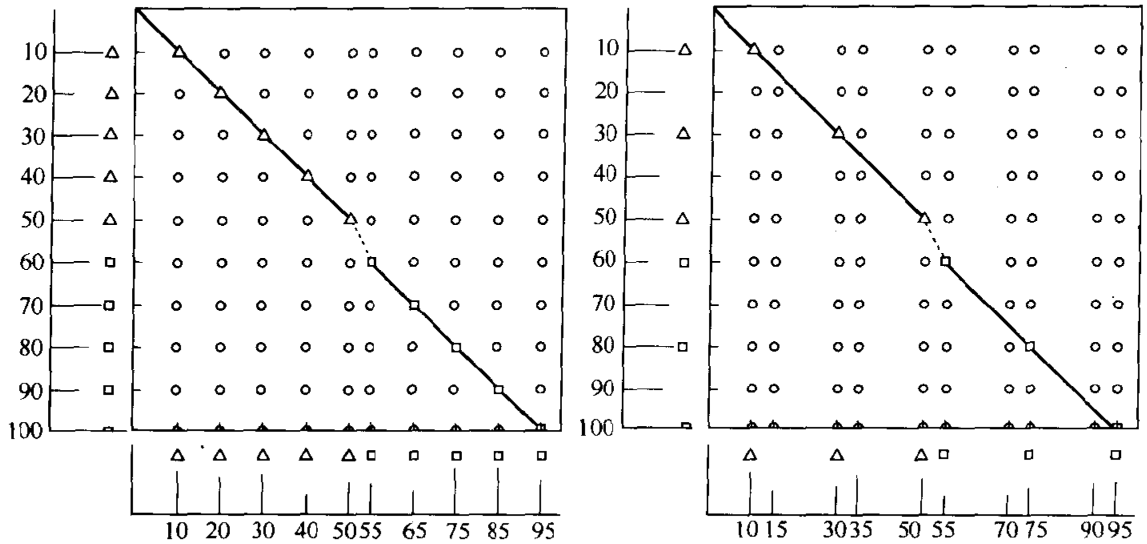


图 11.7 谱 S 可以通过单突变转换为 S' 且 $D(1)=10$（左矩阵）；
谱 S 不能通过单突变转换为 S'' 且 $D(1)=6$（右矩阵）

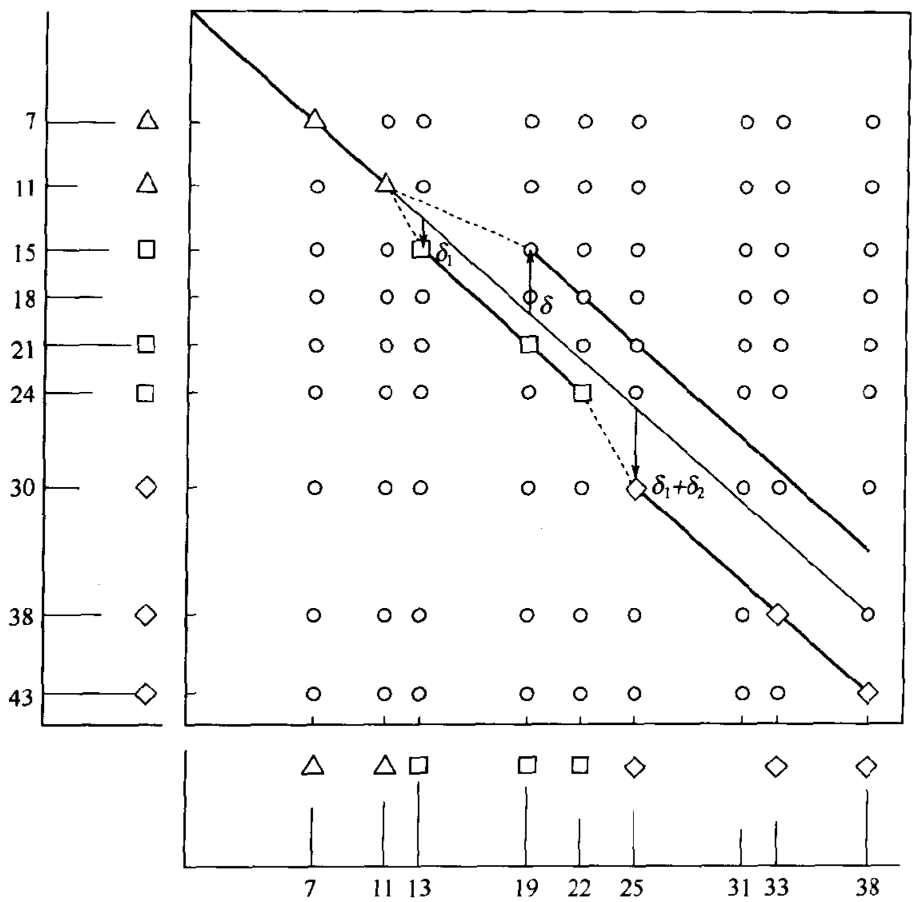


图 11.8 联配谱
共有峰计数揭示主对角线上只有 $D(0)=3$ 个匹配峰，而谱联配可以揭示谱之间更多隐藏的相似性[$D(1)=5$ 和 $D(2)=8$] 并检测到相应的突变

似性概念可以发现 S' 比 S'' 更近似于 S，因为在第一种情形二对角线路径包含了 10 个 1（左矩阵），而在第二种情形则为 6 个 1（右矩阵）。图 11.8 说明随着 k 的增加，谱联配可以检测出更多的谱间弱相似性。下面介绍谱联配的动态规划算法。

令 A_i 和 B_j 分别为 A 的 i-前缀和 B 的 j-前缀。定义 $D_{ij}(k)$ 为 A_i 和 B_j 的 k-相似性，使 A_i 和 B_j 的最后元素可以匹配。换言之，$D_{ij}(k)$ 就是利用至多 $k+1$ 条对角线到达 (a_i, b_j) 的路径中 1 的最大个数。如果 $a_i - a_{i'} = b_j - b_{j'}$，称 (i', j') 和 (i, j) 是联合对角线（co-diagonal）；又如果 $i'<i$，$j'<j$，则 $(i',j')<(i,j)$。考虑到初始条件，引入假象元素 (0, 0)，$D_{0,0}(k)=0$，并设 (0, 0) 是其他任何一个 (i, j) 的联合对角线，那么递归计算 $D_{ij}(k)$ 的动态规划算法为

$$D_{ij}(k) = \max_{(i',j')<(i,j)} \begin{cases} D_{i',j'}(k)+1, & \text{如果}(i',j')\text{和}(i,j)\text{是联合对角线元素} \\ D_{i',j'}(k-1)+1, & \text{其他情况} \end{cases}$$

A 和 B 的 k-相似性为 $D(k)=\max_{ij} D_{ij}(k)$。

上面所描述的谱联配动态规划算法相当慢［对 n 个元素的谱，运算时间为

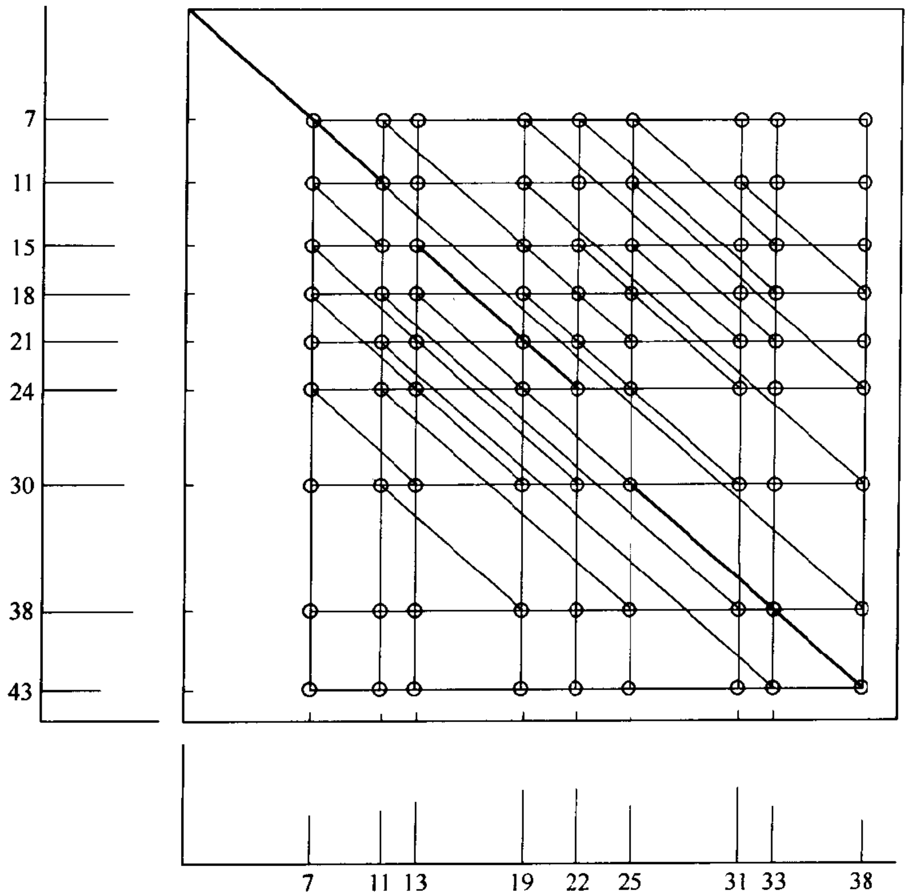


图 11.9 动态规划图的修改可导出快速谱联配算法

$O(n^4k)$]，下面介绍一个求解这个问题的 $O(n^2k)$ 算法。定义 $diag(i,j)$ 作为最大的联合对角线对 (i,j)，使得 $diag(i,j)<(i,j)$。换句话说，$diag(i,j)$ 就是 (a_i, b_j) 所在的对角线上前一个 1 的位置，如果这个位置不存在，则 $diag(i,j)$ 等于 (0,0)。定义

$$M_{ij}(k)=\max_{(i',j')\leqslant(i,j)} D_{i',j'}(k)$$

那么计算 $D_{ij}(k)$ 的递归公式可改写为

$$D_{ij}(k)=\max\begin{cases}D_{diag(i,j)}(k)+1\\M_{i-1,j-1}(k-1)+1\end{cases}$$

计算 $M_{ij}(k)$ 的递归公式是

$$M_{ij}(k)=\max\begin{cases}D_{ij}(k)\\M_{i-1,j}(k)\\M_{i,j-1}(k)\end{cases}$$

上面所描述的动态规划图的转换是通过引入提供对角线转换的水平边和垂直边而完成的（见图 11.9）。路径的得分就是这条路径上的 1 的个数，而 k 对应了转换次数（用到的对角线个数减 1）。

11.10　依靠谱联配肽

上面的简单描述隐藏了许多细节，而正是这些细节给谱联配带来了困难。一个谱通常是一个递增数列（N 末端离子）和递减数列（C 末端离子）的组合。这两个数列形成了谱积 $S\otimes S$ 中的两条对角线，主对角线和垂直对角线，对应于成对 N 末端离子和 C 末端离子。描述的算法没有提到这个细节，只解决主对角线问题。

为了把 N 末端和 C 末端数列组合在一起，考虑$(S_1\cup S_1^R)\otimes(S_2\cup S_2^R)$，其中 S^R 是 P 的反谱。这种变换对每一个 y 离子产生一个 b 版，对每一个 b 离子产生一个 y 版，因此会产生噪声（因为每个噪声峰产生了两次）。另一个更为严重的困难是谱积的每一个 1 都会有一个反向对，在可行的谱联配中，这些反向对中只有一个可以被计数。忽略这个问题可能会得到在反对称路径算法中被剔除的非可行解[79]。

上面所描述的算法也没有包括“依靠谱的序列”比较（“sequence against the spectrum” comparison）情形下的所有相关细节。在这种情形下，动态规划图（见图 11.9）中的水平有向边和垂直有向边受到可能位移的限制，反映了参与突变的氨基酸的质量变化。令 $P=p_1\cdots p_n$ 是用来和谱 $S=\{s_1,\cdots,s_m\}$ 比较的肽。谱S 的 d-前缀包含了 S 中所有 $s_i<d$ 的峰。引入新的变量 $H_{i,d}(k)$，描述肽 P 的 i-前缀 P_i 经过至多 k 个替换到谱 S 的 d-前缀的“最佳”变换。更准确地说，$H_{i,d}(k)$ 等于完全定义的“肽对谱”（peptide versus spectrum）$P\otimes S$ 矩阵

中和对角线（0，0）到（i，d）有 k 位移的最优路径上的 1 的个数。为简单计，假设 P 的理论谱只含有 b 离子。

令 $H_{i,d}(k)$ 是 P_i 经过 k 个替换到 S_d 的“最佳”变换（即，此变换利用了和对角线有至多 k 个位移的、包含 1 的个数为最大的路径）。然而在这种情况下，对角线之间的跳跃并不是任意的，而是由突变氨基酸的质量差异（或相应于化学修饰的质量差异）所限定的。下面描述有替换情况下的动态规划算法（删除、插入和修饰可以类似地递归计算得到）。定义 $x(d)=1$，如果 $d\in S$，而其他情况时 $x(d)=0$。那么 $H_{i,d}(k)$ 有下面的递归公式来描述 [$m(a)$ 表示氨基酸 a 的质量]

$$H_{i,d}(k)=\max\begin{cases}H_{i-1,d-m(p_i)}(k)+x(d)\\ \max_{a=1,20}H_{i,d-[m(a)-m(p_i)]}(k-1)\end{cases}$$

11.11 若干其他的问题和方法

11.11.1 从蛋白质组学到基因组学

质谱技术在鉴定对那些基因包含在序列数据库中的蛋白质的鉴定是非常成功的。然而对串联质谱的从头解释仍旧是一个复杂而耗时的问题，而且事实上质谱技术还没有在发现新基因中起到很大的作用。直到最近 Mann 和 Wilm[231] 还指出他们不能在文献中找到一个关于只利用 MS/MS 的信息来克隆基因的例子。在以后的 5 年里，这种状况有所改变，特别是端粒酶催化亚基的反遗传研究[223]需要 14 个肽的从头测序，并为以后的基因扩增设计 PCR 引物。

11.11.2 大尺度蛋白质分析

复杂的蛋白质混合物可以通过高解析的二维凝胶电泳来分离。分离后，在二维凝胶中每一个点（肽）的本体是未知的，并且必须用质谱来标识。这就需要有效的方法将凝胶中的肽提取出来并将其移入质谱仪中。

12 问　　题

12.1 引言

分子生物学提出了许多有趣的组合问题。Pevzner 和 Waterman[275] 在 1995 年编辑了一本问题集，其中收集了 57 道计算分子生物学中尚未解决的问题。其中 1/4 的问题仅在之后的 5 年中就被解决了。因此不能肯定地说下面的哪个问题还未解决，也许在你读到问题的时候它已经被解决了。

12.2 限制酶切作图

假设一个 DNA 分子被二种限制性内切酶消化了两次。结果片段的间隔图是一个双向间隔图[361]。

问题 12.1　设计一个有效的算法来识别一个双向间隔图。

问题 12.2　设 v 是一个平衡有色图中的偶度（even degree）顶点，$d(v)$ 是边数，与 v 相关。证明 $d(v)$ 可以被分为 $d(v)/2$ 对，使得在同一对中的边有不同的颜色。

设 $P=x_1\cdots x_m$ 是一个 l-色平衡图 $G(V,E)$ 中的一条路。颜色 c 在 P 中是临界的，如果满足①它与 P 中最后一条边（x_{m-1}，x_m）的颜色不同，并且②它是 $E\setminus EP$ 边（与 x_m 关联）中最常出现的颜色（EP 表示路 P 中边的集合）。集合 $E\setminus EP$ 中的边（与 x_m 关联）如果含有临界的颜色则称为临界边。通过每一步选择临界边而得到的路称为临界路。

问题 12.3　说明如何利用临界路径来构建交互欧拉圈。

下面的问题是寻求双色图中 BEST 定理的一种模拟。

问题 12.4　找出一个双色欧拉图中交错欧拉圈的个数。

大多数双酶切（Double Digest）问题的算法都是基于物理图谱中限制片段次序假设的产生和分析。每一个这样的假设对应于一个作图图表 D，它显示了位点的次序和片段的长度（见图 12.1）。注意在作图图表中没有显示位点的坐标，物理图谱 M 提供了关于位点次序和坐标的信息。每一个物理图谱对应于一个图表 $D(M)$（见图 12.1），它标出了对应于位点间距离的片段长度。但反之则不然，不是每一个图表对应于一个物理图谱。这样就出现了一个问题：如何构建一个最近似图表的物理图谱。设 $D=(d_1,\cdots,d_n)$ 和 $M=(m_1,\cdots,m_n)$ 是图表 D 和 $D(M)$ 中所有给定相同次序的片段的长度。图表 D 和图谱 M 间的距离定义为

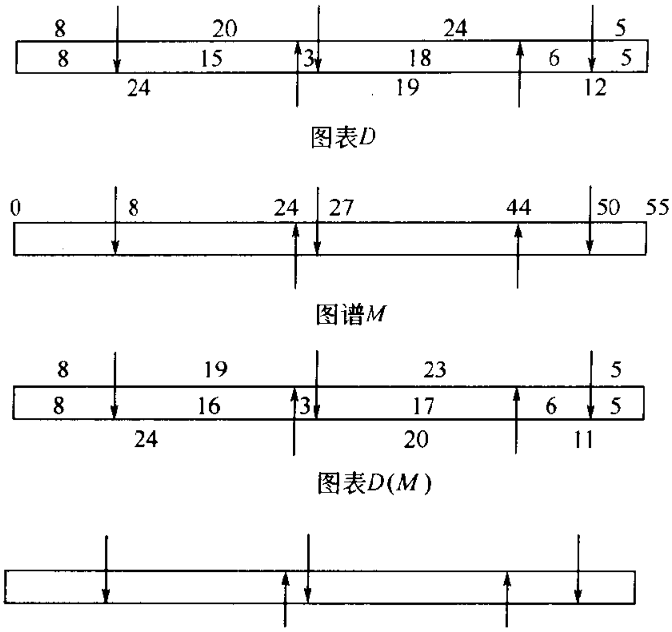


图 12.1 图表和物理图谱

$$d(D,M)=\max_{i=1,n}|d_i-m_i|$$

例如，在图 12.1 中，$D=(8, 20, 24, 5, 24, 19, 12, 8, 15, 3, 18, 6, 5)$，$M=(8, 19, 23, 5, 24, 20, 11, 8, 16, 3, 17, 6, 5)$，则 $d(D, M)=1$。图表调节问题就是找出一个图谱使得它与一个图表有最短距离。

问题 12.5 给定一个图表 D，找出一个图谱 M 使得 $d(D, M)$ 最小。

问题 12.6 给定等价类中的两个图谱，找出最短的一系列盒式变换，将一个图谱变换为另一个图谱。

概括地说，双酶切问题就是有三种酶 A，B 与 C，然后通过单酶切 A，B，C，双酶切 AB，BC，CA，及三重酶切 ABC 实验，从中获得有关片段长度的实验数据。这类实验引出多重酶切问题。

问题 12.7 给定六组实验数据（酶切 A，B，C，AB，BC，CA，及 ABC），找出三种酶 A，B，C 的物理图谱。

问题 12.8 描述多重图谱的盒式变换的特点（三种或更多种酶）。

Rosenblatt 和 Seymour[289] 的 PDP 算法是拟多项式的（pseudo-polynomial）。

问题 12.9 PDP 是否存在多项式算法？

Skiena 等[314] 证明了 n 个元素上的强同效集的最大数 $H(n)$ 是有界的

$$\frac{1}{2}n^{0.6309}\leqslant H(n)\leqslant\frac{1}{2}n^{2.5}$$

其上界似乎相当保守。

问题 12.10 导出 $H(n)$ 更严格的界限。

问题 12.11 证明每一个 5 点集合是可重建的。

问题 12.12 为试探局部酶切问题（probed partial digest problem）设计一个有效的算法。

问题 12.13 导出一个 n 位点加探针部分酶切问题解的最大数的上下界。

光学作图问题的输入是一个元素为 0 或 1 的 $n\times m$ 矩阵 $S=(s_{ij})$，其中每一行对应一个 DNA 分子（正链或负链），每一列对应于这个分子中的一个位置，

如果在分子 i 的 j 位置上有一个切口，则 $s_{ij}=1$。目的是反转分子子集（S 中行的子集）的方向，并找出 t 个列上的“真实切割位点”的一个子集，使得切割位点列上切割位点个数为最大[190]。

解这个问题的一个直接方法是选出包含大部分切割位点的 t 个列，然后找出它们可能的切割位点。但是在这种方法中，每一个真实的位点都将有一个反转的相似位点。设 $w(i, j)$ 是带有切割位点 i 和 j 的分子数目（在正方向或反方向上）。另一种解法是，构建一个顶点为 $\{1, \cdots, m\}$ 的图，两个顶点通过带有权 $w(i, j)$ 的边 (i, j) 相连。

问题 12.14 建立起光学作图和反对称最长路径问题间的联系。

12.3 图谱装配

问题 12.15 求出 n 个克隆的不同交叉的个数。

n 个克隆的交叉可以用一个整数序列 $a_1 \cdots a_n$ 来表示，其中 a_i 为在克隆 i 起始前已终止的克隆的个数。例如，图 3.2 中的 9 个克隆的交叉对应着序列 000112267。但不是每一个整数序列 $a_1 \cdots a_n$ 都表示一个有效的交叉。如果一个探针表示克隆的一次操作 $[i, j]$，这操作蕴涵着不等式 $a_j \leqslant i-1$（克隆 j 和 i 交叠）和 $a_{j+1} \geqslant i-1$（克隆 $i-1$ 和 $j+1$ 不交叠）。

问题 12.16 将给定克隆次序的最短覆盖字符串问题（shortest covering string problem）归结为一个线性整数规划问题并求解。

问题 12.17 记（数据）行上的区间集合为 I，设 I 中两两不相交区间的最大个数为 k。证明在行上存在 k 个点，使得 I 中的每一个区间至少包含其中的一个点。

在一个圆上，对应于弧线集合的交叉图称为圆-弧图。如果一个圆上的弧线集合不能覆盖圆上的某一点 x，则这个弧线集合的圆-弧图为一个区间图（在点 x 处割开圆并将其拉直）。

问题 12.18 设计一个有效的算法来识别圆-弧图。

问题 12.19 若从一个长度为 G 的基因组中选出 N 个长度为 L 的随机克隆，证明用这些克隆表示的基因组的预期分段数大约为 $1-e^c$，其中 $c=\frac{NL}{G}$ 为覆盖度。

在黏粒重叠群作图（cosmid contig mapping）中[376]，克隆交叠信息是由杂交数据产生的。将一组克隆点在膜上进行菌落杂交，用一个已经放射性标记过的克隆来探查膜。这个过程产生交叠信息：哪个探针和别的克隆重叠。如果仅有克隆的一个子集用作探针，那么在不是探针的克隆之间就得不到交叠信息。一个图被称为探针区间图（probe interval graph），如果它的顶点集可被划分成子集 P（作为探针的克隆）和 N（不作为探针的克隆），每一个顶点分配一个区间，使

得两个顶点为相邻的，当且仅当它们相应的区间交叠且至少有一个顶点在 P 中[235]。

问题 12.20 设计一种识别探针区间图的算法。

内积作图（Inner Product Mapping）[263]是一种克隆作图方法，它探查一组辐射杂交克隆两次，一次用 BAC，一次用 STS，获得与 STS 有关的 BAC 图谱。内积作图需要两组数据：一组是用 STS 的杂交筛选矩阵，另一组是用 BAC 的杂交筛选矩阵。

问题 12.21 给定用 STS 和 BAC 的杂交筛选矩阵，构建一个与 STS 有关的 BAC 图谱。

在排列 $\pi=\pi_1\cdots\pi_n$ 中，元素 $\pi_i\pi_j\pi_k(1\leqslant i<j<k\leqslant n)$ 构成一个有序三联体。设 $\Phi(\pi)$ 为 π 的所有 $\binom{n}{3}$ 个有序三联体的集合，辐射杂交作图引出了下列问题。

问题 12.22 给定一个 n 元素集合的有序三联体的任意集合 T，找出一个排列 π 使得 $T\subset\Phi(\pi)$。

如果 $1\leqslant i<j<k\leqslant n$ 或 $1\leqslant k<j<i\leqslant n$，元素 $\pi_i\pi_j\pi_k$ 构成一个无序三联体。设 $\Theta(\pi)$ 为 π 的所有无序三联体的集合。

问题 12.23 给定一个 n 元素集合的无序三联体的任意集合 T，找出一个排列 π 使得 $T\subset\Theta(\phi)$。

12.4 测序

解决最短超字符串问题（Shortest Superstring Problem，SSP）的最简单的启发式方法是 GREEDY 算法：不断合并带有最大交叠的一对字符串，直到还剩一个字符串。Tarhio 和 Ukkonen[333]将 SSP 算法中的压缩（compression）定义为这个算法所保留的符号个数与简单连接的所有字符串之比。

问题 12.24 证明 GREEDY 算法对一个最佳超字符串至少达到 1/2 的压缩，即

$$\frac{\text{GREEDY 压缩}}{\text{最佳压缩}}\geqslant\frac{1}{2}$$

关于压缩的性能保证并不表示关于长度的性能保证。因为还没有找到 GREEDY 近似比小于 2 的例子，Blum 等[37]提出了下列假设。

问题 12.25 证明 GREEDY 的性能保证达到 2。

设 $\mathcal{S}=\{s_1,\cdots,s_n\}$ 是一个线状字符串集，$\mathcal{C}=\{c_1,\cdots,c_m\}$ 是一个环状字符串集。如果每一个 s_i 包含在某一个环状字符串 c_j 中（$1\leqslant j\leqslant m$），称 $\mathcal{C}$ 为 $\mathcal{S}$ 的一个循环。循环的长度 $|\mathcal{C}|=\sum_{j=1}^{m}|c_j|$ 为来自 $\mathcal{C}$ 的字符串的全部长度。

问题 12.26 对于一个线状字符串集找出其最短循环。

设 $P=\{s_1,\cdots,s_m\}$ 为一组正字符串，$N=\{t_1,\cdots,t_k\}$ 为一组负字符串。假设没有负字符串 t_i 是任何一个正字符串 s_j 的子串。字符串 s 是（P，N）的一个一致超字符串，使得每一个 s_i 是 s 的一个子串且没有 t_i 是 s 的子串[180]。

问题 12.27 为最短一致超字符串问题设计一种近似算法。

测序中读出的短片段含有错误，使得片段装配复杂化。读入错误导致了最短 k-近似超字符串问题（shortest k-approximate superstring problem）[181]。

问题 12.28 给定一个字符串集 $\mathcal{S}$，找出最短字符串 w，使得 $\mathcal{S}$ 中的每一个字符串 x 与 w 的某一子字符串相匹配且最多有 k 个错配。

假定在 A-字符字母表上给定一个长度固定为 l 的 n 个随机字符串的集合 $\mathcal{S}$。若 n 很大（阶数为 A^l），则集合 $\mathcal{S}$ 的最短公共超字符串的长度 $E(\mathcal{S})$ 的阶数为 n。若 n 很小，则 $E(\mathcal{S})$ 的阶数为 nl。

问题 12.29 将 $E(\mathcal{S})$ 表示为 l，n 和 A 的函数。

给定字符串 s 和 t，则 overlap(s，t) 为 s 匹配 t 后缀的 s 最大前缀的长度。

问题 12.30 给定一组固定长度的 i. i. d.（独立同分布）字符串 $\{s_1, \cdots, s_n\}$，求出 $\max\{\text{overlap}(s_i,s_j): 1\leqslant i\neq j\leqslant n\}$ 的分布。

给定 DNA 测序项目中的一组读序 $\mathcal{S}=\{s_1,\cdots,s_n\}$ 和一个整数 l，$\mathcal{S}$ 的谱是来自字符串 $s_1,\cdots,s_n$ 的所有 l 元组的集合 $\mathcal{S}_l$。在每个 DNA 读序中，设误差数目的上界为 Δ。片段装配问题的一种解法是，首先纠正每个读序的错误，然后将正确的读序装配成重叠群。这引出了下面的问题。

问题 12.31 给定 $\mathcal{S}$，Δ 和 l，在 $\mathcal{S}$ 中的每个字符串中引入最多 Δ 个修正，使得 $|\mathcal{S}_l|$ 为最小。

12.5 DNA 阵列

问题 12.32 证明 明确地重构长度为 n 的任意字符串所需探针的数目其信息理论（information-theoretic）上的下界为 $\Omega(n)$。

问题 12.33 设计一个利用带误差（假阳性和假阴性）数据的 SBH（杂交测序）序列重构算法。

给定两个带有相同 l 元组组成的字符串，它们之间的距离定义为将一个字符串转换为另一个时所需的最短系列调换的长度。

问题 12.34 设计一种算法来计算或近似带有相同 l 元组组成的两个字符串之间的距离。

问题 12.35 带有相同 l 元组组成的两个 n 字符的字符串之间的最大距离是多少?

连续堆积杂交（continuous stacking hybridization）假定短探针的一次附加杂交连续地延伸了由目标 DNA 片段和测序阵列中的探针构成的双螺旋。在此方

法中，在阵列 $C(k)$ 上带有一个短 m 元组的附加杂交提供了有关某些包含在序列中的（$k+m$）元组的信息。

问题 12.36 给定谱 S，它不能提供明确的 SBH 重构，试确定明确地重构目标片段所需的连续堆积杂交实验的最小次数。

问题 12.37 给定谱 S 和附加连续堆积杂交的结果，重构一个 DNA 片段的序列。

一个 l 阶的简化二元阵列是一个带有 2×2^l 存储量的、由所有两类多探针

$$\underbrace{\{W,S\},\{W,S\},\cdots,\{W,S\}}_{l}, \quad 与 \underbrace{\{R,Y\}\{R,Y\},\cdots,\{R,Y\}}_{l}$$

组成的阵列。

例如，当 $l=2$ 时，简化二元阵列由 8 个多探针组成：WW，WS，SW，SS，RR，RY，YR 和 YY。每一个多探针是四种二核苷酸的组合。

问题 12.38 计算简化二元阵列的分支概率，并将其与均匀阵列的分支概率作比较。

如果阵列中所有探针的长度最大为 k，则称该阵列为 k-有界阵列。

问题 12.39 给定 m 和 k，找出一个带有 m 个（多）探针的 k-有界阵列，使其分辨能力最大。

上述问题的一个简易形式如下。

问题 12.40 在所有的 k-有界阵列中，二元阵列 $C_{\text{bin}}(k-1)$ 是否渐近地近似最佳分辨率？

尽管二元阵列比均匀阵列提供更好的分辨率，但如何从一个二元阵列上的 DNA 片段谱来重构 DNA 片段尚无有效的算法。

问题 12.41 是否存在一个可利用二元阵列进行 SBH 序列重构的多项式算法？

定理 5.6 的证明考虑了 x，y 处的转换 s_{i+1} 和 z，u 处的转换 s_i。在该证明中隐含了顶点集 $\{x, y\}$ 和 $\{z, u\}$ 没有交叠的假设。

问题 12.42 调整证明使其适应顶点集交叠的情形。

设 Λ 是一组字符串。考虑由不同 l 元组的所有有序对组成的集合 E（字符串先后次序数据），使得它们以任意距离及给定的顺序在 Λ 中的一些字符串中出现。Chetverin 和 Kramer[66]提出了 DNA 阵列的嵌套链杂交方法，它引出了下列问题[291]。

问题 12.43 给定字符串先后次序数据重构 Λ。

对于最小化均匀 DNA 阵列的边界长度，二维 Gray 码是最佳的。但是对于一个任意阵列，光刻掩模的全部边界长度的最小化问题还未解决。

问题 12.44 在一个（任意）阵列中找出探针的一种配置，使得光刻阵列设计的掩模的全部边界为最小。

12.6 序列比较

将序列 V 拟合进序列 W 就是这样一个问题：找出 W 的一条子字符串 W'，使得联配 $s(V, W')$ 的得分在所有 W 的子字符串中为最大。

问题 12.45 设计拟合问题的一种有效算法。

问题 12.46 估计一下在两条 n 字符序列之间不同联配的数目。

问题 12.47 设计一种算法来计算一对字符串间不同的最优联配的数目。

问题 12.48 对于一对字符串 $v_1, \cdots, v_n$ 和 $w_1, \cdots, w_m$，对每一个 (i, j) 说明如何计算字符 v_i 和字符 w_j 联配的最佳联配值。

问题 12.49 给定参数 k，计算两个字符串间的全局联配，使服从联配中最多包含 k 个缺口（连续插删空格）的约束。

k-差异联配问题就是找出字符串 V 和 W 的最佳全局联配，最多包含 k 个误配、插入或缺失。

问题 12.50 设计一个 $O(kn)$ 的 k-差异全局联配算法来比较两个 n 字符字符串。

Chao 等在 1992 年[64]描述了一种在对角形带（diagonal band）中联配两条序列的算法，该算法只需要 $O(nw)$ 的计算时间和 $O(n)$ 的（贮存）空间，其中 n 是序列的长度，w 是带的宽度。

问题 12.51 在对角形带内，一个算法可否在 $O(w)$（贮存）空间内实现?

Myers 和 Miller[246]研究了下列问题。

问题 12.52 发展一种带仿射缺口罚分（affine gap penalties）的全局序列联配算法，其所需（贮存）空间是线性的。

Huang 和 Miller[169]研究了下列问题。

问题 12.53 发展一种局部联配算法，其所需（贮存）空间是线性的。

若用空间有效的方法来解序列联配问题，则一个大小为 $n\times m$ 的原问题可被简化为两个大小为 $i\times\frac{m}{2}$ 和 $(n-i)\times\frac{m}{2}$ 的子问题。在快速平行地执行序列联配时，需要有一个均衡划分（balanced partitioning)，使得原问题被分解为大小相同的子问题。

问题 12.54 设计一种具有均衡划分的空间有效的联配算法。

局部联配的得分不能通过匹配区域的长度来标准化。这样，将会选择长度为 100、得分为 100 的局部联配，而不选择长度为 10、得分为 99 的局部联配，虽然后者也许在生物学上更有意义。为了在记分中反映局部联配的长度，将包含子字符串 I 和 J 的局部联配的得分 $s(I,J)$ 调整为 $\frac{s(I,J)}{|I|+|J|}$，$|I|+|J|$ 为联配区域的总长度。标准化局部联配（normalized local alignment）问题就是找出子字符

串 I 和 J，使得在所有子字符串 I 和 J 中 $\frac{s(I,J)}{|I|+|J|}$ 为最大，且 $|I|+|J|\geqslant k$，其中 k 为 I 和 J 的最小长度的阈值。

问题 12.55 设计一种求解标准化局部联配问题的算法。

如果 V 是 X 的子序列，字符串 X 称为字符串 V 的超序列。

问题 12.56 给定字符串 V 和 W，设计一种寻找 V 和 W 的最短超序列的算法。

设 P 是长度为 n 的（特征序列）模式，T 是长度为 m 的文本。串联重复(tandem repeat）问题就是在 T 中找出一个区段，该区段与 P 的某个串联重复有最佳全局联配。设 P^m 为 P 自身连接 m 次的多联体，则串联重复问题等价于计算 P^m 和 T 之间的局部联配，而解这个问题的标准局部联配算法需要的时间为 $O(nm^2)$。

问题 12.57 找出一种求解串联重复问题的方法，其（计算）时间为 $O(nm)$。

环状字符串联配的定义为：将这些环状字符串在任意位置切开（使其直线化），然后再联配这些线形字符串。

问题 12.58 找出环状字符串的最优联配（局部和全局）。

两条不同字符串 A 和 B 间的局部联配就是找一对子字符串，一条在 A 中，另一条在 B 中，使它们具有最大相似度。假设希望在字符串 A 中找出一对（不重叠的）子字符串，使它们具有最大相似度（最优不严格重复问题）。在 A 与其自身之间计算局部联配并不解决问题，因为联配的结果也许和重叠的子字符串一致。Miller（未发表的手稿）以及后来的 Kannan 和 Myers[184] 与 Schmidt[308] 研究了这个问题。

问题 12.59 设计一种处理最优不严格重复问题的算法。

Schoniger 和 Waterman[310] 扩展了序列联配中编辑操作的范围，在插入、删除和替换之外还包括了非重叠反转（non-overlapping reversal)。

问题 12.60 设计一种带非重叠反转的有效的序列联配算法。

嵌合联配（chimeric alignment）问题[205] 就是给定一个字符串 V 和字符串的数据库 $\mathcal{W}=\{W_1, \cdots, W_N\}$，要找出 $\max\limits_{1\leqslant i\neq j\leqslant N} s(V, W_i\oplus W_j)$，其中 $W_i\oplus W_j$ 表示 W_i 和 W_j 的串联。

问题 12.61 设计一种求解嵌合联配问题的有效算法。

问题 12.62 证明在 n 个不同整数的任意排列中一定存在一个子序列，它要么是一个长度至少为 $\sqrt{n}$ 的递增子序列，要么为一个长度至少为 $\sqrt{n}$ 的递减子序列。

一个 Catalan（卡塔兰）序列是 n 个 1 和 n 个 0 的排列 $x_1, \cdots, x_{2n}$，使得对任意的前缀 $x_1, \cdots, x_i$，1 的个数至少和 0 的个数一样多。第 n 个 Catalan 数 C_n 是这类序列的个数。

问题 12.63 证明下列命题。

- C_n 为带两行长度为 n 的标准杨氏表（Young tableau）的个数。
- C_n 是带有长度最大为 2 的最长递减子序列的排列 $\pi \in S_n$ 的数目。
- C_n 是正整数序列 $1 \leqslant a_1 \leqslant a_2 \leqslant \cdots \leqslant a_n$ 的个数，且对所有 i 有 $a_i \leqslant i$。

问题 12.64 证明递归式 $C_{n+1} = C_n C_0 + C_{n-1} C_1 + \cdots + C_0 C_n$ 。

问题 12.65 证明排列 π 的最长递减子序列的长度为杨氏表 $P(\pi)$ 的第一列的长度。

排列 π 的一个子序列 σ 为 k 递增的，如果作为一个集合，它可以记为

$$\sigma = \sigma_1 \cup \sigma_2 \cup \cdots \cup \sigma_k$$

其中任意给定的 σ_i 是排列 π 的一个递增子序列。

问题 12.66 设计一个算法来找出最长的 k 递增子序列。

对于 n 个字符的 i. i. d. 字符串 V 和 W 间的编辑距离 $d(V, W)$ 这种情况，Chang 和 Lampe[61]提出了类似 Sankoff-Mainville 猜想的一种想法。

问题 12.67 对于 k 个字符字母表中的随机的 i. i. d. 的 n 个字符的字符串，证明

$$\frac{Expectation(d(V,W))}{n} = 1 - \frac{1}{\sqrt{k}} + o\left(\frac{1}{\sqrt{k}}\right)$$

Gusfield 等[146]证明了对于全局联配，参数空间分解（parameter space decomposition）中凸多边形的个数囿于 $O(n^{2/3})$，其中 n 是序列的长度。Fernández-Baca等[102]研究了下面的问题。

问题 12.68 生成一对长度为 n 的序列，使得它们在参数空间分解中有一个 $\Omega(n^{2/3})$阶的区域。

对于一个固定长度的字母表，在参数空间分解中具有 $\Omega(n^{2/3})$ 区域的序列的例子还是未知的。

问题 12.69 在有界字母表的情况下，对全局联配，改进空间分解中区域个数的界限 $O(n^{2/3})$。

问题 12.70 对两条长度为 n 的随机序列的全局联配，导出空间分解中区域期望个数的界限。

问题 12.71 在多重联配的情况下推导空间分解中区域个数的界限。

局部联配的参数空间分解所包含的区域通常比全局联配的参数空间分解所包含的区域多。Vingron 和 Waterman[346]研究了参数序列联配与相变间的联系。在此联系中，研究对数域中的参数分解是有趣的。

问题 12.72 对于两条长度为 n 的随机序列的局部联配，导出对数域的空间分解中区域期望个数的界限。

处理两条序列的参数序列联配的 Gusfield 等[146]算法每区域的运行时间为 $O(R+E)$，其中 R 是参数分解中区域的个数，E 是执行一个单联配所需的时间。在非加权记分方法情形，$R=O(E)$，所以每个区域耗时为 $O(E)$。当使用加权矩

阵时，对 R 所知甚少。Gusfield 提出了下列问题。

问题 12.73 在加权矩阵的情况下，估计凸分解中区域的个数。

问题 12.74 在加权矩阵的情况下，设计一个空间分解的快速算法。

因为 RNA 折叠的能量参数的估计带有误差，所以研究参数 RNA 折叠是有用的。例如，通过比较对应于三叶草状 tRNA 参数空间分解的区域，可得到现行 RNA 能量参数精确度的估计。

问题 12.75 发展一种参数 RNA 折叠的算法。

设 $S_n(\mu, \sigma)$ 是相应于两条长度为 n 的随机 i. i. d. 字符串间全局联配得分（#匹配-μ#不匹配-σ#插删）的随机变量。Arratia 和 Waterman[14] 定义了 $a(\mu, \delta)=\lim\limits_{n\to\infty}\frac{S_n(\mu,\delta)}{n}$，并证明了 $\{a=0\}=\{(\mu,\delta):a(\mu,\delta)=0\}$ 是一连续相变曲线。

问题 12.76 描述曲线 $a(\mu, \delta)=0$ 的特征。

12.7 多重联配

问题 12.77 设计一个多重联配的空间有效的算法。

问题 12.78 设计一种算法，由三种方式联配（3-way alignments）来装配多重联配。

问题 12.79 构建一个例子，使得 Vingron-Argos 矩阵乘法算法需要 $\Omega(L)$ 步迭代，其中 L 是序列长度。

Jiang 和 Li[180] 阐述了下列问题。

问题 12.80 二元字母表上的最短共有超序列（SCS）和最长共有超序列（LCS）能否被好于 2 的比率来近似?

也许有人会争论说 NP 完全性的概念对一些计算生物学问题稍微有些误导，因为它对实际中很有用的有限参数区间不敏感。例如，在许多应用中，对于解决 $k\leqslant 10$ 的多重联配的有效算法很感兴趣。现在所用的是 $O((2n)^k)$ 动态规划算法。如果将注意力集中在 $k\leqslant 10$ 的范围，多重联配问题的 NP-完全性几乎不能提供值得期待的东西。对每一个 k 的固定值，它甚至会出现具有线性时间算法这种情况。例如，如果问题可以在 $O(2^k n)$ 时间里被解决，它将和 NP 完全性完全一致。

最近十年发展的一些算法对固定参数范围的多重联配这类问题特别有用。现在还不知道通过动态规划得到的复杂度是否就是多重联配问题复杂度的“最新发现”[38]。Mike Fellows 提出下列猜想。

问题 12.81 对于固定元数的字母表中的 k 条序列，其最长共有子序列问题可在 $f(k)n^\alpha$ 时间内解决，其中 α 与 k 无关。

12.8 探测 DNA 中的信号

问题 12.82 描述 B 在投掷硬币打赌中成功的策略。

问题 12.83 描述 A 在投掷硬币打赌中最佳策略（即，使损失最小的策略）。

如果在投掷硬币打赌中的硬币有偏倚［例如 $p(0)>p(1)$］，则 A 选择如 0…0这样的结果对提高胜利的概率是有意义的。

问题 12.84 研究在投掷硬币打赌中的硬币有偏倚的情况下，B 对 A 而言是否还具有优势？

问题 12.85 在线性字符串情况下，对一个给定的词导出其出现次数的方差。

问题 12.86 设计一个共有词问题的近似算法。

解码问题可被归结为有向无圈图中的最长路问题。这就引出了一个问题，是关于 Viterbi 算法的空间有效的形式。

问题 12.87 解码问题是否存在一个空间为线性的算法？

12.9 基因预测

剪接联配算法（spliced alignment algorithm）通过将相关蛋白质作为模板在基因组 DNA 中寻找外显子。如果模板是别的（未注释的）基因组 DNA 而不是蛋白质，那么会怎么样呢？尤其是，是否可以用（未注释的）小鼠基因组 DNA 来预测人类基因？

问题 12.88 归结剪接联配算法，将一个基因组序列和另一个基因组序列联配。

问题 12.89 在可以利用多重相似蛋白质的情况下，归结基于相似性的方法来进行基因预测。

Sze 和 Pevzner[332] 对下列问题做了简洁陈述。

问题 12.90 修改剪接联配算法来找出次最优剪接联配。

“说谎者与 20 个问题的游戏”假设游戏中每个答案错误的概率为 p。显然，如果 $p=\frac{1}{2}$，则游戏是无意义的，因为说谎者没有向我们传递任何信息。

问题 12.91 为“说谎者与 20 个问题的游戏”设计一个有效的策略，若错误答案的概率 $p\neq\frac{1}{2}$，从区间$[1, n]$ 找出 k 个未知整数。

问题 12.92 若错误答案的概率 $p\neq\frac{1}{2}$，估计“说谎者与 20 个问题的游戏”中问题的期望数。

问题 12.93 对于给定的基因组序列，设计实验和计算流程来找出所有可选择的剪接变量。基于 PCR 的查询可以用来测试可能有指数个数的剪接变量的假设，此观察引出了上述问题的另一种形式。

问题 12.94 给定带（未知）路径集 $\mathcal{C}$ 的图 $G(V, E)$，通过最少次数的提问："集 $\mathcal{C}$ 中是否包括通过 G 中的顶点 v 和 w 的路?"来重构 $\mathcal{C}$。

设 S 是一组固定的探针，C 是一个 DNA 序列。C 的指纹是与 C 杂交的 S 中的一部分探针。设 G 是一个基因组序列，它包含由克隆 C 的 cDNA 表示的基因。Mironov 和 Pevzner[241]研究了下列基于指纹的基因识别问题。

问题 12.95 给定基因组序列 G 和相应于克隆 C 的 cDNA 指纹，预测 G 中的基因（即预测 G 中所有的外显子）。

12.10 基因组重排

反序排序（sorting by reversal）可以消除断点。然而对于某些排列（如563412），没有反序可以减少断点。563412 中所有的三条带（无断点的最大间距）都是递增的。

问题 12.96 证明 如果一个无符号排列有一个递减带，则存在至少减少一个断点的反序。

反序排序 π 的 2-贪婪（2-greedy）算法选择反序 ρ 和 σ，使得 $\pi \cdot \rho \cdot \sigma$ 中的断点在所有反序对中为最小。

问题 12.97 证明 2-贪婪算法是反序排序的一个保证性能的算法。

在基因序列包含重复的情形，是用反序将排序排列（permutation）变换为用反序排序词（word）。例如，将 43132143 变换为 42341314 的最短反序序列包括两次反转：43132143→42313143→42341314。

问题 12.98 为利用反序排序词设计一个保证性能的算法。

Kececioglu 和 Sankoff[193]研究了在有符号排列情形下直径 $D(n)$ 的界限，并证明了 $n-1 \leqslant D(n) \leqslant n$。他们提出了下列猜想。

问题 12.99 对于有符号环形排列，当 n 足够大时 $D(n) = n$。

问题 12.100 证明对于 n 个元素的"难以排序"的有符号排列，$d(\pi) = D(n)$。

问题 12.101 对于预期的反序距离，改进下界并导出一个上界。

问题 12.102 估计反序距离的方差。

Gates 和 Papadimitriou[120]猜想：对一特定的 n 个元素的排列，其排序至少需要 $\frac{19}{16}n$ 次反序。Heydari 和 Sudborough[161]反驳了他们的猜测，并用 $\frac{18}{16}n+2$ 次反序排序 Gates-Papadimitriou 的排列。

问题 12.103 找出对称组的前缀反序直径（prefix reversal diameter）。

基因组的进化不仅通过反转（inversion），也通过对换（transposition）。对于一个排列 π，对换 $\rho(i, j, k)$（定义对所有 $1 \leqslant i < j \leqslant n+1$ 和 $1 \leqslant k \leqslant n+1$，使

得 $k\notin[i,\ j]$）在 π_{k-1} 和 π_k 之间“插入”一个 π 的区间 $[i,\ j-1]$。于是，$\rho(i,\ j,\ k)$ 对应于排列

$$\begin{pmatrix} 1\cdots i-1 & \boxed{i\ i+1\cdots\ \cdots\ \cdots j-2\ j-1} & \boxed{j\cdots k-1} & k\cdots n \\ 1\cdots i-1 & \boxed{j\cdots k-1} & \boxed{i\ i+1\cdots\ \cdots\ \cdots j-2\ j-1} & k\cdots n \end{pmatrix}$$

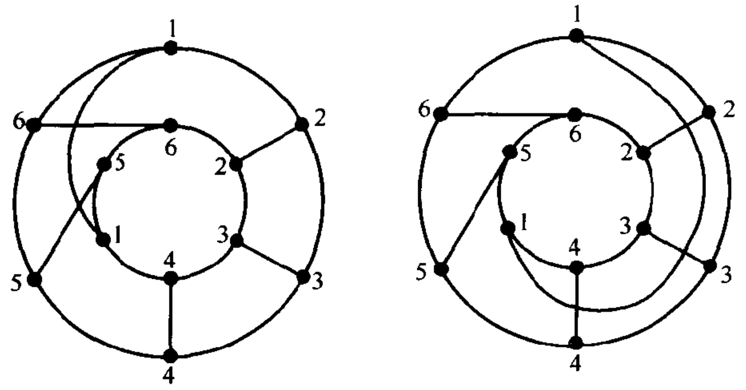
给定排列 π 和 σ，对换距离是将 π 变换为 $\pi\cdot\rho_1\cdot\rho_2\cdots\rho_t=\sigma$ 的一系列变换 ρ_1，ρ_2，…，ρ_t 中最短系列的长度。利用对换来排序 π，就是找出 π 和恒等排列 l 间的对换距离 $d(\pi)$。Bafna 和 Pevzner[20]设计了一个 1.5 性能保证的算法来通过对换进行排序，并证明了对称组的对换直径 $D_t(n)$ 的界限为 $\frac{n}{2}\leqslant D_t(n)\leqslant\frac{3n}{4}$。

问题 12.104 找出对称组的对换直径。

利用对换来排序的有名的变异体是通过对换 $\rho(i,\ i+1,\ i+2)$ 来排序，其中的操作为交换邻近的元素。对于线性排列而言，这问题有一种简单的冒泡-排序（bubble-sort）算法来解决。对于环状排列，这问题的解决就更难。

问题 12.105 对于环状排列设计一种算法，通过交换邻近元素来排序。

问题 12.106 每个环状排列可通过 $2\left\lceil\frac{n-1}{2}\right\rceil\left\lfloor\frac{n-1}{2}\right\rfloor$ 次交换邻近元素来排序。



(a) (b)

图 12.2 嵌入中的交叉边

用均匀分布在环上的元素 $\pi_1\cdots\pi_n$ 来表示环状排列。图 12.2 显示了位于两个同心圆上的环状排列 $\pi=\pi_1\cdots\pi_n$ 和 $\sigma=\sigma_1\cdots\sigma_n$，在 n 条边 $e_1\cdots e_n$ 中，每条边 e_i 将 π 中元素 i 和 σ 中元素 i 相连。将 π 和 σ 这种表示方法称为嵌入（embedding）。感兴趣的是，使交叉边个数 C 最小的嵌入。嵌入中边的方向或为顺时针、或为逆时针；注意到交叉边的总数依赖于方向的选择。例如，图 12.2（a）中的嵌入对应于 $C=2$，而图 12.2（b）中的嵌入对应于 $C=3$。一个 n 元方向向量 $v=v_1$，…，v_n，其中 $v_i\in\{+1,\ -1\}$，定义了一个嵌入。若 $v_i=+1$，则边 e_i 为顺时针，否则边 e_i 为逆时针。

为方便起见，在一个环上选择“十二点钟”顶点作为环状排列的“起始”点。选择元素 r 作为 π 的起始点来定义 π 的旋转。为简单计，假设 $\sigma=1\cdots n$ 是恒等排列，且 σ 的起始点为 1。

π 的所有以 r 为起始点的旋转和所有向量 $\boldsymbol{v}$ 定义了一个带 $C(r,\boldsymbol{v})$ 条交叉边的嵌入。Sankoff 和 Goldstein 在 1989 年[303]研究了下列最优嵌入问题。

问题 12.107 找出$\min\limits_{r,\boldsymbol{v}} C(r,\boldsymbol{v})$。

设 $d_{ij}(\pi)$ 为排列 π 中元素 i 到元素 j 逆时针的距离。一条顺时针边 e_i 的长度定义为 $\hat{l}=(d_{1i}(\sigma)-d_{1i}(\pi))\mathrm{mod}(n)$，而一条逆时针边的长度定义为 $i=n-\hat{i}$。对于一个旋转 r，根据规则：当顺时针边 e_i 比逆时针边 e_i 短时（即，若 $\hat{i}<\check{i}$）$v_i(r)=+1$，反之 $v_i(r)=-1$，定义规范方向向量 $\boldsymbol{v}(r)=[v_1(r)\cdots v_n(r)]$。

问题 12.108 证明 $\min C[r,\boldsymbol{v}(r)]\leqslant\left\lceil\frac{n-1}{2}\right\rceil\left\lfloor\frac{n-1}{2}\right\rfloor$。

问题 12.109 证明：对于每个 $0\leqslant r\leqslant n-1$，环状排列 π 最多经过 $C[r,\boldsymbol{v}(r)]$ 次交换邻近元素可被排序。

n 个元素的 $\mathcal{A}$-排列是指一个排列 $\{1,2,\cdots,n\}$ 中散布着字符表 $\mathcal{A}$ 中的字符。例如，3aa1baa4a52b 是一个带 $\mathcal{A}=\{a,b\}$ 的 31452 的 $\mathcal{A}$-排列。当一个 $\mathcal{A}$-排列的起始元素和末尾元素重合时，它的反序是有效的。一个排列称为恒等 $\mathcal{A}$-排列，其中 $\{1,2,\cdots,n\}$ 的元素是按次序出现的（$\mathcal{A}$ 中的元素是任意分配的）。

问题 12.110 设计一种测试，判断一个 $\mathcal{A}$-排列是否可通过有效反序来排序。

问题 12.111 设计一种算法，通过有效反序来排序 $\mathcal{A}$-排列。

12.11 计算蛋白质组学

质谱法已成为新蛋白质序列的一个来源，其中一些序列先前在 DNA 层次上是未知的。这就引出了如何将蛋白质组学和基因组学之间的缺口连接的问题，即以从大规模 MS/MS 肽测序中得到的信息为基础测序 DNA 的问题。因为通过质谱法来测序的肽会分为短的片段，并带有潜在的模糊性（如邻近氨基酸的转换和通配符），所以这个问题是复杂的。

问题 12.112 根据一给定的蛋白质设定一组肽（带有模糊性），设计实验和计算方案来找出相应于此蛋白质的基因组序列。

问题 12.113 设计一个算法，从蛋白质数据库中搜索带模糊性的肽。

考虑一个可被蛋白酶（如胰岛素）完全消化的蛋白质的混合物。这是一个长度为 10～20 个氨基酸的肽的集合，现在的问题是确定哪些肽属于同一个蛋白质，并在每个蛋白质中重构肽的次序。质谱法可以（部分）排序混合物中的所有肽，但哪些肽属于同一蛋白质以及蛋白质中肽的次序还是未知的。蛋白质混合物的蛋

白质测序是将肽装配为单个蛋白质的问题。

问题 12.114 设计实验和计算方案，通过质谱法来测序蛋白质混合物。

肽从头测序的谱图方法没有将内部和多重荷电离子考虑在内。

问题 12.115 设计一个考虑内部和多重荷电离子的肽测序算法。

设 $M(P)$ 是肽 P 的所有肽段的质量的集合。利用不同的非特定的蛋白酶消化 P，可以获得一组实验测得的肽段的质量 $M \subset M(P)$。

问题 12.116 给定一组肽段的质量 $M \subset M(P)$，重构 P。

正确决定肽母体质量在从头肽测序中是至关重要的。母体质量中的一个错误会导致 C 末端离子顶点质量的系统错误，进而使肽重构变得困难。在实际中，真的肽质量（由肽的所有氨基酸给出）和实验观察到的母体质量之间的偏差通常很大，使得肽重构中的误差几乎是不可避免。

问题 12.117 给定一个 MS/MS 谱（无母体质量），设计一个算法来估计母体质量。

13 必须了解的分子生物学

关于分子生物学的知识，建议看 Lewin 在 1999 年写的书[220]。下面对必须了解的内容作一简要的介绍。

DNA 是由核苷酸 A、T、G 和 C 四种字母组成的字符串❶。一个生物体的完整的 DNA 称为该生物体的基因组。生物体（例如人）是由数万亿个细胞组成的，而每个细胞含有相同的基因组。DNA 的长度从几百万个核苷酸（细菌）到几十亿个核苷酸（哺乳动物）不等。DNA 形成一个螺旋，但这对于本书来说其实并不重要。更重要的是 DNA 通常是以双链形式存在，一条链和另一条链以 Watson-Crick 方式互补（T 与 A 配对，C 和 G 配对），例如

```
A T G C T C A G G
| | | | | | | | |
T A C G A G T C C
```

DNA 制造出细胞的骨干被称之为蛋白质。蛋白质是由表示 20 种氨基酸的字母所组成的短的字符串。人类基因组大约编码 100000 种蛋白质，每种蛋白质长约几百个氨基酸。细菌编码 500～1500 种蛋白，这个数量接近于一个生物体存活所需的最低限度。蛋白质是由被称为基因的 DNA 片段编码出来的，基因的长度大约是它对应的蛋白质的三倍。为什么是三倍呢？因为在 DNA 中每三个核苷酸编码蛋白质中的一个氨基酸。因此有 $4^3=64$ 个三元组（密码子），这就有一个问题，为什么自然界需要这么多组合来编码 20 个氨基酸。遗传密码（见图 13.1）是冗余的，更不用说表示蛋白质末端的终止密码子了。

生物学家将生物界划分为真核生物（DNA 被围在细胞核内）和原核生物。原核生物的基因组只有一条链，而真核生物的基因组有一组链，称之为染色体组。需要记住在原核生物和真核生物之间的主要区别，就是在原核生物中，基因是连续的，没有中断的；而在真核生物中，基因往往断裂为一组片段（称之为外显子）。人的基因有的可被断裂成多达 50 个外显子，这些外显子被看似无意义的叫作内含子的片段分隔开。

一个被断裂成很多段的基因仍能产生相应的蛋白质。为了实现这个目标，细胞必须切掉内含子，并将所有的外显子连接起来。这是通过一个类似于短的单链 DNA 的中间分子 mRNA 来实现的，这个过程称为转录。在 DNA 上有起始转录的信号，被称为启动子。然后，蛋白质合成机构将 mRNA 中的密码子翻译成氨

❶ 这是从生物信息学角度而言的通俗说法——译者注。

基酸的字符串（蛋白质）。在实验室里，mRNA 也能作为模板来产生一个称为 cDNA 的互补拷贝。cDNA 与切去内含子的原始基因相同。

第二位

第一位	T		C		A		G	
T	TTT	PHE	TCT	SER	TAT	TYR	TGT	CYS
	TTC		TCC		TAC		TGC	
	TTA	LEU	TCA		TAA	终止	TGA	终止
	TTG		TCG		TAG		TGG	TRP
C	CTT	LEU	CCT	PRO	CAT	HIS	CGT	ARG
	CTC		CCC		CAC		CGC	
	CTA		CCA		CAA	GLN	CGA	
	CTG		CCG		CAG		CGG	
A	ATT	ILE	ACT	THR	AAT	ASN	AGT	SER
	ATC		ACC		AAC		AGC	
	ATA		ACA		AAA	LYS	AGA	ARG
	ATG	MET	ACG		AAG		AGG	
G	GTT	VAL	GCT	ALA	GAT	ASP	GGT	GLY
	GTC		GCC		GAC		GGC	
	GTA		GCA		GAA	GLU	GGA	
	GTG		GCG		GAG		GGG	

图 13.1 遗传密码

这些年来，生物学家已经学会了如何用 DNA 制造很多东西。他们也学会了如何大量地复制 DNA 用于进一步的研究。其中的一个方法是 PCR（polymerase chain reaction，聚合酶链式反应），它就像 DNA 的古腾堡印刷机。PCR 扩增一个短的（100～500 个核苷酸）DNA 片段，能产生大量相同的片段。要使用 PCR，就必须知道感兴趣区域两端的短（20～30 个核苷酸）序列，并设计出两个 PCR 引物。PCR 引物是与这些序列相同的合成的 DNA 片段。为什么需要大量短的相同的 DNA 片段？从计算科学的观点看，有 10^{18}拷贝的相同的字符串并不算多，它并不能增加信息量。但是对于生物学家来说这意味着很多，因为大多数生物实验需要用很多序列。例如，PCR 可以被用来检测一个 DNA 样品中某一个 DNA 片段的存在。

另一个复制 DNA 的方法是克隆。与 PCR 不同的是，克隆不需要关于两侧引物的任何先验信息。然而在克隆中，生物学家并不能控制何种 DNA 片段被扩增。这个过程通常先将 DNA 打断成小的片段。为了研究单独的片段，生物学家通过片段克隆得到每个片段的很多相同的拷贝。克隆将一个 DNA 片段合并到一个克隆载体中。克隆载体（通常源于一个病毒或高等生物的 DNA）是一个能让

另一个 DNA 片段插入的 DNA 分子。在这个操作中，克隆载体不会失去其自我复制的能力。载体将外源 DNA 导入宿主细胞中（例如细菌），在那里它们能大量地增殖。这种自我复制过程产生出这个片段的大量拷贝，使它的结构得以研究。用这种方式复制的片段叫克隆。生物学家从同一个 DNA 分子中，能制备出由上千个克隆（每个克隆代表一个短的、随机选择的 DNA 片段）所组成的克隆文库。

限制酶是在每一个确定的词出现的位置上剪切 DNA 的分子剪刀。例如，*Bam*H I限制酶在每个 GGATCC 出现的位置上将 DNA 切成限制片段。蛋白质也能被另一类叫蛋白酶的剪切成短的片段（称为多肽）。

将两个互补的 DNA 链结合为一个双链 DNA 分子的过程称为杂交。用一个与已知 DNA 片段互补的短的探针的杂交能用来检测这个 DNA 片段的存在与否。探针是一个短的荧光标记的单链 DNA 片段，用来检测在 DNA 样品中是否有一个互补的序列存在。为什么需要荧光标记的探针呢？如果一个探针可以与一个 DNA 片段杂交，那么就能用光谱检测仪检测到信号。

凝胶电泳是一种允许生物学家不用对 DNA 片段测序就能测量 DNA 片段大小的技术。DNA 是一个携带负电荷的分子，在电场中能向正极迁移。迁移的速度与片段的大小有关，因此测量迁移的距离就能使生物学家估计出 DNA 片段的大小。

参考文献

[1] A.V. Aho and M.J. Corasick. Efficient string matching: an aid to bibliographic search. *Communication of ACM,* 18:333–340, 1975.

[2] D.J. Aldous and P. Diaconis. Hammersley's interacting particle process and longest increasing subsequences. *Probability Theory and Related Fields,* 103:199–213, 1995.

[3] F. Alizadeh, R.M. Karp, L.A. Newberg, and D.K. Weisser. Physical mapping of chromosomes: A combinatorial problem in molecular biology. *Algorithmica,* 13:52–76, 1995.

[4] F. Alizadeh, R.M. Karp, D.K. Weisser, and G. Zweig. Physical mapping of chromosomes using unique probes. *Journal of Computational Biology,* 2:159–184, 1995.

[5] S. Altschul, W. Gish, W. Miller, E. Myers, and J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology,* 215:403–410, 1990.

[6] S.F. Altschul. Amino acid substitution matrices from an information theoretic perspective. *Journal of Molecular Biology,* 219:555–565, 1991.

[7] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped Blast and Psi-Blast: a new generation of protein database search programs. *Nucleic Acids Research,*25:3389–3402, 1997.

[8] T.S. Anantharaman, B. Mishra, and D. C. Schwartz. Genomics via optical mapping. II: Ordered restriction maps. *Journal of Computational Biology,* 4:91–118, 1997.

[9] A. Apostolico. Improving the worst-case performance of the Hunt-Szymanski strategy for the longest common subsequence of two strings. *Information Processing Letters,* 23:63–69, 1986.

[10] A. Apostolico and F. Preparata. Data structures and algorithms for the string statistics problem. *Algorithmica,* 15: 481–494, 1996.

[11] R. Arratia, E.S. Lander, S. Tavare, and M.S. Waterman. Genomic mapping by anchoring random clones: a mathematical analysis. *Genomics,* 11:806–827, 1991.

[12] R. Arratia, D. Martin, G. Reinert, and M.S. Waterman. Poisson process approximation for sequence repeats, and sequencing by hybridization. *Journal of Computational Biology,* 3:425–464, 1996.

[13] R. Arratia and M.S. Waterman. The Erdös-Rényi strong law for pattern matching with a given proportion of mismatches. *Annals of Probability,* 17:1152–1169, 1989.

[14] R. Arratia and M.S. Waterman. A phase transition for the score in matching random sequences allowing deletions. *Annals of Applied Probability,* 4:200–225, 1994.

[15] R. Baer and P. Brock. Natural sorting over permutation spaces. *Math. Comp.,* 22:385–410, 1968.

[16] R.A. Baeza-Yates and G.H. Gonnet. A new approach to text searching. In *Proceedings of the Twelfth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pages 168–175, Cambridge, Massachussets, 1989.

[17] R.A. Baeza-Yates and C.H. Perleberg. Fast and practical approximate string matching. In *Third Annual Symposium on Combinatorial Pattern Matching,* volume 644 of *Lecture Notes in Computer Science,* pages 185–192, Tucson, Arizona, April/May 1992. Springer-Verlag.

[18] V. Bafna, E.L. Lawler, and P.A. Pevzner. Approximation algorithms for multiple sequence alignment. *Theoretical Computer Science,* 182:233–244, 1997.

[19] V. Bafna and P.A. Pevzner. Genome rearrangements and sorting by reversals. *SIAM Journal on Computing,* 25:272–289, 1996.

[20] V. Bafna and P.A.Pevzner. Sorting by transpositions. *SIAM Journal on Discrete Mathematics*, 11:224–240, 1998.

[21] J. Baik, P.A. Deift, and K. Johansson. On the distribution of the length of the longest subsequence of random permutations. *Journal of the American Mathematical Society*, 12:1119–1178, 1999.

[22] W. Bains. Multan: a program to align multiple DNA sequences. *Nucleic Acids Research*, 14:159–177, 1986.

[23] W. Bains and G. Smith. A novel method for nucleic acid sequence determination. *Journal of Theoretical Biology*, 135:303–307, 1988.

[24] P. Baidi and S. Brunak. *Bioinformatics: The Machine Learning Approach*. The MIT Press, 1997.

[25] E. Barillot, B. Lacroix, and D. Cohen. Theoretical analysis of library screening using an N-dimensional pooling strategy. *Nucleic Acids Research*, 19:6241–6247, 1991.

[26] C. Bartels. Fast algorithm for peptide sequencing by mass spectroscopy. *Biomedical and Environmental Mass Spectrometry*, 19:363–368, 1990.

[27] G.J. Barton and M.J.E. Sternberg. A strategy for the rapid multiple alignment of protein sequences. *Journal of Molecular Biology*, 198:327–337, 1987.

[28] A. Baxevanis and B.F. Ouellette. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*. Wiley-Interscience, 1998.

[29] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[30] G. Benson. Sequence alignment with tandem duplication. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, *Proceedings of the First Annual International Conference on Computational Molecular Biology (RECOMB-97)*, pages 27–36, Santa Fe, New Mexico, January 1997. ACM Press.

[31] G. Benson. An algorithm for finding tandem repeats of unspecified pattern size. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, *Proceedings of the Second Annual International Conference on Computational Molecular Biology (RECOMB-98)*, pages 20–29, New York, New York, March 1998. ACM Press.

[32] S.M. Berget, C. Moore, and P.A. Sharp. Spliced segments at the 5' terminus of adenovirus 2 late mRNA. *Proceedings of the National Academy of Sciences USA*, 74:3171–3175, 1977.

[33] P. Berman and S. Hannenhalli. Fast sorting by reversal. In *Seventh Annual Symposium on Combinatorial Pattern Matching*, volume 1075 of *Lecture Notes in Computer Science*, pages 168–185, Laguna Beach, California, June 1996. Springer-Verlag.

[34] P. Berman, Z. Zhang, Y.I. Wolf, E.V. Koonin, and W. Miller. Winnowing sequences from a database search. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, *Proceedings of the Third Annual International Conference on Computational Molecular Biology (RECOMB-99)*, pages 50–58, New York, New York, March 1999. ACM press.

[35] K. Biemann and H.A. Scoble. Characterization of tandem mass spectrometry of structural modifications in proteins. *Science*, 237:992–998, 1987.

[36] B.E. Blaisdell. A measure of the similarity of sets of sequences not requiring sequence alignment. *Proceedings of the National Academy of Sciences USA*, 16:5169–5174, 1988.

[37] A. Blum, T. Jiang, M. Li, J. Tromp, and M. Yannakakis. Linear approximation of shortest superstrings. *Journal of the ACM*, 41:630–647, 1994.

[38] H.L. Bodlaender, R.G. Downey, M.R. Fellows, and H.T. Wareham. The parameterized complexity of sequence alignment and consensus. *Theoretical Computer Science*, 147:31–54, 1995.

[39] M. Boehnke, K. Lange, and D.R. Cox. Statistical methods for multipoint radiation hybrid mapping. *American Journal of Human Genetics*, 49:1174–1188, 1991.

[40] K.S. Booth and G.S. Leuker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13:335–379, 1976.

[41] P. Bork and T.J. Gibson. Applying motif and profile searches. *Methods in Enzymology*, 266:162–184, 1996.

[42] M. Borodovsky and J. McIninch. Recognition of genes in DNA sequences with ambiguities. *BioSystems*, 30:161–171, 1993.

[43] M. Yu. Borodovsky, Yu.A. Sprizhitsky, E.I. Golovanov, and A.A. Alexandrov. Statistical features in the *E. coli*

genome functional domains primary structure III. Computer recognition of protein coding regions. *Molecular Biology*, 20:1144–1150, 1986.

[44] D. Botstein, R.L. White, M. Skolnick, and R.W. Davis. Construction of a genetic linkage map in man using restriction fragment length polymorphisms. *American Journal of Human Genetics*, 32:314–331, 1980.

[45] R.S. Boyer and J.S. Moore. A fast string searching algorithm. *Communication of ACM*, 20:762–772, 1977.

[46] A. Brazma, I. Jonassen, I. Eidhammer, and D. Gilbert. Approaches to the automatic discovery of patterns in biosequences. *Journal of Computational Biology*, 5:279-305, 1998.

[47] V. Brendel, J.S. Beckman, and E.N. Trifonov. Linguistics of nucleotide sequences: morphology and comparison of vocabularies. *Journal of Biomolecular Structure and Dynamics*, 4:11–21, 1986.

[48] D. Breslauer, T. Jiang, and Z. Jiang. Rotations of periodic strings and short superstrings. *Journal of Algorithms*, 24:340–353, 1997.

[49] N. Broude, T. Sano, C. Smith, and C. Cantor. Enhanced DNA sequencing by hybridization. *Proceedings of the National Academy of Sciences USA*, 91:3072–3076, 1994.

[50] S. Brunak, J. Engelbrecht, and S. Knudsen. Prediction of human mRNA donor and acceptor sites from the DNA sequence. *Journal of Molecular Biology*, 220:49–65, 1991.

[51] W.J. Bruno, E. Knill, D.J. Balding, D.C. Bruce, N.A. Doggett, W.W. Sawhill, R.L. Stallings, C.C. Whittaker, and D.C. Torney. Efficient pooling designs for library screening. *Genomics*, 26:21–30, 1995.

[52] R. Bundschuh and T. Hwa. An analytic study of the phase transition line in local sequence alignment with gaps. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, *Proceedings of the Third Annual International Conference on Computational Molecular Biology (RECOMB-99)*, pages 70–76, Lyon, France, April 1999. ACM Press.

[53] C. Burge, A.M. Campbell, and S. Karlin. Over- and under-representation of short oligonucleotides in DNA sequences. *Proceedings of the National Academy of Sciences USA*, 89:1358–1362, 1992.

[54] C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 268:78–94, 1997.

[55] S. Burkhardt, A. Crauser, P. Ferragina, H.-P. Lenhof, E. Rivals, and M. Vingron. q-gram based database searching using a suffix array. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, *Proceedings of the Third Annual International Conference on Computational Molecular Biology (RECOMB-99)*, pages 77–83, Lyon, France, April 1999. ACM Press.

[56] A. Caprara. Formulations and complexity of multiple sorting by reversals. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, *Proceedings of the Third Annual International Conference on Computational Molecular Biology (RECOMB-99)*, pages 84–93, Lyon, France, April 1999. ACM Press.

[57] A. Caprara. Sorting by reversals is difficult. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, *Proceedings of the First Annual International Conference on Computational Molecular Biology (RECOMB-97)*, pages 75–83, Santa Fe, New Mexico, January 1997. ACM Press.

[58] H. Carrillo and D. Lipman. The multiple sequence alignment problem in biology. *SIAM Journal on Applied Mathematics*, 48:1073–1082, 1988.

[59] R.P. Carstens, J.V. Eaton, H.R. Krigman, P.J. Walther, and M.A. Garcia Blanco. Alternative splicing of fibroblast growth factor receptor 2 (FGF-R2) in human prostate cancer. *Oncogene*, 15:3059–3065, 1997.

[60] W.K. Cavenee, M.F. Hansen, M. Nordenskjold, E. Kock, I. Maumenee, J.A. Squire, R.A. Phillips, and B.L. Gallie. Genetic origin of mutations predisposing to retinoblastoma. *Science*, 228:501–503, 1985.

[61] W.I. Chang and J. Lampe. Theoretical and empirical comparisons of approximate string matching algorithms. In *Third Annual Symposium on Combinatorial Pattern Matching*, volume 644 of *Lecture Notes in Computer Science*, pages 175–184, Tucson, Arizona, April/May 1992. Springer-Verlag.

[62] W.I. Chang and E.L. Lawler. Sublinear approximate string matching and biological applications. *Algorithmica*, 12:327–344, 1994.

[63] K.M. Chao. Computing all suboptimal alignments in linear space. In *Fifth Annual Symposium on Combinatorial Pattern Matching*, volume 807 of *Lecture Notes in Computer Science*, pages 31–42, Asilomar, California, 1994. Springer-Verlag.

[64] K.M. Chao, W.R. Pearson, and W. Miller. Aligning two sequences within a specified diagonal band. *Computer*

Applications in Biosciences, 8:481–487, 1992.

[65] M. Chee, R. Yang, E. Hubbel, A. Berno, X.C. Huang, D. Stern, J. Winkler, D.J. Lockhart, M.S. Morris, and S.P.A. Fodor. Accessing genetic information with high density DNA arrays. *Science*, 274:610–614, 1996.

[66] A. Chetverin and F. Kramer. Sequencing of pools of nucleic acids on oligonucleotide arrays. *BioSystems*, 30:215–232, 1993.

[67] L.T. Chow, R.E. Gelinas, T.R. Broker, and R.J. Roberts. An amazing sequence arrangement at the 5' ends of adenovirus 2 messenger RNA. *Cell*, 12:1–8, 1977.

[68] I. Chumakov, P. Rigault, S. Guillou, P. Ougen A. Billaut, G. Guasconi, P. Gervy, I. LeGall, P. Soularue, and L. Grinas et al. Continuum of overlapping clones spanning the entire human chromosome 21q. *Nature*, 359:380–387, 1992.

[69] G. Churchill. Stochastic models for heterogeneous DNA sequences. *Bulletin of Mathematical Biology*, 51:79–94, 1989.

[70] V. Chvátal and D. Sankoff. Longest common subsequences of two random sequences. *Journal of Applied Probability*, 12:306–315, 1975.

[71] V. Chvatal and D. Sankoff. An upper-bound techniques for lengths of common subsequences. In D. Sankoff and J.B. Kruskal, editors, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, pages 353–357. Addison-Wesley, 1983.

[72] K.R. Clauser, P.R. Baker, and A.L. Burlingame. The role of accurate mass measurement (+/– 10ppm) in protein identification strategies employing MS or MS/MS and database searching. *Analytical Chemistry*, 71:2871–2882, 1999.

[73] F.S. Collins, M.L. Drumm, J.L. Cole, W.K. Lockwood, G.F. Vande Woude,and M.C. Iannuzzi. Construction of a general human chromosome jumping library, with application to cystic fibrosis. *Science*, 235:1046–1049, 1987.

[74] N.G. Copeland, N.A. Jenkins, D.J. Gilbert, J.T. Eppig, L.J. Maltals, J.C.Miller, W.F. Dietrich, A. Weaver, S.E. Lincoln, R.G. Steen, L.D. Steen, J.H. Nadeau, and E.S. Lander. A genetic linkage map of the mouse: Current applications and future prospects. *Science*, 262:57–65, 1993.

[75] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. The MIT Press, 1989.

[76] A. Coulson, J. Sulston, S. Brenner, and J. Karn. Toward a physical map of the genome of the nematode,*Caenorhabditis elegans*. *Proceedings of the National Academy of Sciences USA*, 83:7821–7825, 1986.

[77] D.R. Cox, M. Burmeister, E.R. Price, S. Kim, and R.M. Myers. Radiation hybrid mapping: a somatic cell genetic method for constructing highresolution maps of mammalian chromosomes. *Science*, 250:245–250, 1990.

[78] E. Czabarka, G. Konjevod, M. Marathe, A. Percus, and D.C. Torney. Algorithms for optimizing production DNA sequencing. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2000)*, pages 399–408, San Francisco, California, 2000. SIAM Press.

[79] V. Dancik, T. Addona, K. Clauser, J. Vath, and P.A. Pevzner. De novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, 6:327–342, 1999.

[80] K.J. Danna, G.H. Sack, and D. Nathans. Studies of simian virus 40 DNA. Ⅶ. a cleavage map of the SV40 genome. *Journal of Molecular Biology*, 78:263–276, 1973.

[81] K.E. Davies, P.L. Pearson, P.S. Harper, J.M. Murray, T. O'Brien, M. Sarfarazi, and R. Williamson. Linkage analysis of two cloned DNA sequences flanking the Duchenne muscular dystrophy locus on the short arm of the human X chromosome. *Nucleic Acids Research*, 11:2303–2312, 1983.

[82] M.A. Dayhoff, R.M. Schwartz, and B.C. Orcutt. A model of evolutionary change in proteins. In *Atlas of Protein Sequence and Structure*, chapter 5, pages 345–352. 1978.

[83] J. Deken. Some limit results for longest common subsequences. *Discrete Mathematics*, 26:17–31, 1979.

[84] J. Deken. Probabilistic behavior of longest common subsequence length. In D. Sankoff and J.B. Kruskal, editors, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, pages 359–362. Addison-Wesley, 1983.

[85] A. Dembo and S. Karlin. Strong limit theorem of empirical functions for large exceedances of partial sums of i.i.d. variables. *Annals of Probability*, 19:1737–1755, 1991.

[86] R.P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, 51:161–165, 1950.

[87] T. Dobzhansky and A.H. Sturtevant. Inversions in the chromosomes of *Drosophila pseudoobscura*. *Genetics*, 23:28–64, 1938.

[88] H. Donis-Keller, P. Green, C. Helms, S. Cartinhour, B. Weiffenbach, K. Stephens, T.P. Keith, D.W. Bowden, D.R. Smith, and E.S. Lander. A genetic linkage map of the human genome. *Cell*, 51:319–337, 1987.

[89] R.F. Doolittle, M.W. Hunkapiller, L.E. Hood, S.G. Devare, K.C. Robbins, S.A. Aaronson, and H.N. Antoniades. Simian sarcoma virus onc gene, vsis, is derived from the gene (or genes) encoding a platelet-derived growth factor. *Science*, 221:275–277, 1983.

[90] R. Drmanac, S. Drmanac, Z. Strezoska, T. Paunesku, I. Labat, M. Zeremski, J. Snoddy, W. K. Funkhouser, B. Koop, and L. Hood. DNA sequence determination by hybridization: a strategy for efficient large-scale sequencing. *Science*, 260:1649–1652, 1993.

[91] R. Drmanac, I. Labat, I. Brukner, and R. Crkvenjakov. Sequencing of megabase plus DNA by hybridization: theory of the method. *Genomics*, 4:114–128, 1989.

[92] J. Dumas and J. Ninio. Efficient algorithms for folding and comparing nucleic acid sequences. *Nucleic Acids Research*, 10:197–206, 1982.

[93] R. Durbin, S. Eddy, A. Krogh, and G. Mitchinson. *Biological Sequence Analysis*. Cambridge University Press, 1998.

[94] M. Dyer, A. Frieze, and S. Suen. The probability of unique solutions of sequencing by hybridization. *Journal of Computational Biology*, 1:105–110, 1994.

[95] S.R. Eddy and R. Durbin. RNA sequence analysis using covariance models. *Nucleic Acids Research*, 22:2079–2088, 1994.

[96] N. El-Mabrouk, D. Bryant, and D. Sankoff. Reconstructing the pre-doubling genome. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, *Proceedings of the Third Annual International Conference on Computational Molecular Biology (RECOMB-99)*, pages 154–163, Lyon, France, April 1999. ACM Press.

[97] J. Eng, A. McCormack, and J. Yates. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of American Society for Mass Spectometry*, 5:976–989, 1994.

[98] G.A. Evans and K.A. Lewis. Physical mapping of complex genomes by cosmid multiplex analysis. *Proceedings of the National Academy of Sciences USA*, 86:5030–5034, 1989.

[99] W. Feldman and P.A. Pevzner. Gray code masks for sequencing by hybridization. *Genomics*, 23:233–235, 1994.

[100] D. Feng and R. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, 25:351–360, 1987.

[101] D. Fenyo, J. Qin, and B.T. Chait. Protein identification using mass spectrometric information. *Electrophoresis*, 19:998–1005, 1998.

[102] D. Fernandez-Baca, T. Seppalainen, and G. Slutzki. Bounds for parametric sequence comparison. In *Sixth International Symposium on String Processing and Information Retrieval*, pages 55–62, Cancun, Mexico, September 1999. IEEE Computer Society.

[103] J. Fernández-de Cossío, J. Gonzales, and V. Besada. A computer program to aid the sequencing of peptides in collision-activated decomposition experiments. *Computer Applications in Biosciences*, 11:427–434, 1995.

[104] J.W. Fickett. Recognition of protein coding regions in DNA sequences. *Nucleic Acids Research*, 10: 5303–5318, 1982.

[105] J.W. Fickett. Finding genes by computer: the state of the art. *Trends in Genetics*, 12:316–320, 1996.

[106] J.W. Fickett and C.S. Tung. Assessment of protein coding measures. *Nucleic Acids Research*, 20:6441–6450, 1992.

[107] W.M. Fitch and T.F. Smith. Optimal sequence alignments. *Proceedings of the National Academy of Sciences USA*, 80:1382–1386, 1983.

[108] H. Fleischner. *Eulerian Graphs and Related Topics*. Elsevier Science Publishers, 1990.

[109] S.P.A. Fodor, R.P. Rava, X.C. Huang, A.C. Pease, C.P. Holmes, and C.L. Adams. Multiplex biochemical assays with biological chips. *Nature*, 364:555–556, 1993.

[110] S.P.A. Fodor, J.L. Read, M.S. Pirrung, L. Stryer, A.T. Lu, and D. Solas. Light-directed spatially addressable parallel chemical synthesis. *Science*, 251:767–773, 1991.

[111] S. Foote, D. Vollrath, A. Hilton, and D.C. Page. The human Y chromosome: overlapping DNA clones spanning the euchromatic region. *Science*, 258:60–66, 1992.

[112] D. Fousler and S. Karlin. Maximum success duration for a semi-markov process. *Stochastic Processes and their Applications*, 24:203–210, 1987.

[113] D. Frishman, A. Mironov, H.W. Mewes, and M.S. Gelfand. Combining diverse evidence for gene recognition in completely sequenced bacterial genomes. *Nucleic Acids Research*, 26:2941–2947, 1998.

[114] D.R. Fulkerson and O.A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15: 835–856, 1965.

[115] D.J. Galas, M. Eggert, and M.S. Waterman. Rigorous pattern-recognition methods for DNA sequences. Analysis of promoter sequences from Escherichia coli. *Journal of Molecular Biology*, 186:117–128, 1985.

[116] Z. Galil and R. Giancarlo. Speeding up dynamic programming with applications to molecular biology. *Theoretical Computer Science*, 64:107–118, 1989.

[117] J. Gallant, D. Maier, and J. Storer. On finding minimal length superstrings. *Journal of Computer and System Science*, 20:50–58, 1980.

[118] M. Gardner. On the paradoxial situations that arise from nontransitive relationships. *Scientific American*, pages 120–125, October 1974.

[119] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., 1979.

[120] W.H. Gates and C.H. Papadimitriou. Bounds for sorting by prefix reversals. *Discrete Mathematics*, 27:47–57, 1979.

[121] M.S. Gelfand. Computer prediction of exon-intron structure of mammalian pre-mRNAs. *Nucleic Acids Research*, 18:5865–5869, 1990.

[122] M.S. Gelfand. Statistical analysis and prediction of the exonic structure of human genes. *Journal of Molecular Evolution*, 35:239–252, 1992.

[123] M.S. Gelfand. Prediction of function in DNA sequence analysis. *Journal of Computational Biology*, 2:87–115, 1995.

[124] M.S. Gelfand and E. V. Koonin. Avoidance of palindromic words in bacterial and archaeal genomes: a close connection with restriction enzymes. *Nucleic Acids Research*, 25:2430–2439, 1997.

[125] M.S. Gelfand, A.A. Mironov, and P.A. Pevzner. Gene recognition via spliced sequence alignment. *Proceedings of the National Academy of Sciences USA*, 93:9061–9066, 1996.

[126] J.F. Gentleman and R.C. Mullin. The distribution of the frequency of occurrence of nucleotide subsequences, based on their overlap capability. *Biometrics*, 45:35–52, 1989.

[127] W. Gillett, J. Daues, L. Hanks, and R. Capra. Fragment collapsing and splitting while assembling high-resolution restriction maps. *Journal of Computational Biology*, 2:185–205, 1995.

[128] P.C. Gilmore and A.J. Hoffman. A characterization of comparability graphs and of interval graphs. *Canadian Journal of Mathematics*, 16:539–548, 1964.

[129] W. Gish and D.J. States. Identification of protein coding regions by database similarity search. *Nature Genetics*, 3:266–272, 1993.

[130] L. Goldstein and M.S. Waterman. Mapping DNA by stochastic relaxation. *Advances in Applied Mathematics*, 8:194–207, 1987.

[131] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander. Molecular classification of cancer:class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[132] M. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, 1980.

[133] G.H. Gonnet, M.A. Cohen, and S.A. Benner. Exhaustive matching of the entire protein sequence database. *Science*, 256:1443–1445, 1992.

[134] A. Gooley and N. Packer. The importance of co- and post-translational modifications in proteome projects. In W. Wilkins, K. Williams, R. Appel, and D. Hochstrasser, editors, *Proteome Research: New Frontiers in Functional Genomics*, pages 65–91. Springer-Verlag, 1997.

[135] O. Gotoh. Consistency of optimal sequence alignments. *Bulletin of Mathematical Biology*, 52:509–525, 1990.

[136] P. Green. Documentation for phrap. http://bozeman.mbt.washington.edu/phrap.docs/phrap.html.

[137] D.S. Greenberg and S. Istrail. Physical mapping by STS hybridization: algorithmic strategies and the challenge of software evaluation. *Journal of Computational Biology*, 2:219–273, 1995.

[138] M. Gribskov, J. Devereux, and R.R. Burgess. The codon preference plot: graphic analysis of protein coding sequences and prediction of gene expression. *Nucleic Acids Research*, 12:539–549, 1984.

[139] M. Gribskov, M. McLachlan, and D. Eisenberg. Profile analysis: detection of distantly related proteins. *Proceedings of the National Academy of Sciences USA*, 84:4355–4358, 1987.

[140] R. Grossi and F. Luccio. Simple and efficient string matching with k mismatches. *Information Processing Letters*, 33:113–120, 1989.

[141] L.J. Guibas and A.M. Odlyzko. String overlaps, pattern matching and nontransitive games. *Journal of Combinatorial Theory, Series A*, 30:183–208, 1981.

[142] R. Guigo, S. Knudsen, N. Drake, and T.F. Smith. Prediction of gene structure. *Journal of Molecular Biology*, 226:141–157, 1992.

[143] J.F. Gusella, N.S. Wexler, P.M. Conneally, S.L. Naylor, M.A. Anderson, R.E. Tanzi, P.C. Watkins, K. Ottina, M.R. Wallace, A.Y. Sakaguchi, A.B. Young, I. Shoulson, E. Bonilla, and J.B. Martin. A polymorphic DNA marker genetically linked to Huntington's disease. *Nature*, 306:234–238, 1983.

[144] D. Gusfield. Efficient methods for multiple sequence alignment with guaranteed error bounds. *Bulletin of Mathematical Biology*, 55:141–154, 1993.

[145] D. Gusfield. *Algorithms on Strings, Trees, and Sequences. Computer Science and Computational Biology*. Cambridge University Press, 1997.

[146] D. Gusfield, K. Balasubramanian, and D. Naor. Parametric optimization of sequence alignment. *Algorithmica*, 12:312–326, 1994.

[147] D. Gusfield, R. Karp, L. Wang, and P. Stelling. Graph traversals, genes and matroids: An efficient case of the travelling salesman problem. *Discrete Applied Mathematics*, 88:167–180, 1998.

[148] J.G. Hacia, J.B. Fan, O. Ryder, L. Jin, K. Edgemon, G. Ghandour, R.A. Mayer, B. Sun, L. Hsie, C.M. Robbins, L.C. Brody, D. Wang, E.S. Lander, R. Lipshutz, S.P. Fodor, and F.S. Collins. Determination of ancestral alleles for human single-nucleotide polymorphisms using high-density oligonucleotide arrays. *Nature Genetics*, 22:164–167, 1999.

[149] C.W. Hamm, W.E. Wilson, and D.J. Harvan. Peptide sequencing program. *Computer Applications in Biosciences*, 2:115–118, 1986.

[150] J.M. Hammersley. A few seedlings of research. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statististics and Probabilities*, pages 345-394, Berkeley, California, 1972.

[151] S. Hannenhalli. Polynomial algorithm for computing translocation distance between genomes. In *Sixth Annual Symposium on Combinatorial Pattern Matching*, volume 937 of *Lecture Notes in Computer Science*, pages 162–176, Helsinki, Finland, June 1995. Springer-Verlag.

[152] S. Hannenhalli, C. Chappey, E. Koonin, and P.A. Pevzner. Genome sequence comparison and scenarios for gene rearrangements: A test case. *Genomics*, 30: 299–311, 1995.

[153] S. Hannenhalli and P.A. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, pages 581–592, Milwaukee, Wisconsin, 1995.

[154] S. Hannenhalli and P.A. Pevzner. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). In *Proceedings of the 27th Annual ACM Symposium on the Theory of Computing*, pages 178–189, 1995 (full version appeared in Journal of ACM, 46: 1–27, 1999).

[155] S. Hannenhalli and P.A. Pevzner. To cut ... or not to cut (applications of comparative physical maps in molecular evolution). In *Seventh Anuual ACM-SIAM Symposium on Discrete Algorithms*, pages 304–313, Atlanta, Georgia, 1996.

[156] S. Hannenhalli, P.A. Pevzner, H. Lewis, S. Skeina, and W. Feldman. Positional sequencing by hybridization. *Computer Applications in Biosciences*, 12:19–24, 1996.

[157] W.S. Hayes and M. Borodovsky. How to interpret an anonymous bacterial genome: machine learning approach to gene identification. *Genome Research*, 8:1154–1171, 1998.

[158] S. Henikoff and J.G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences USA*, 89:10915–10919, 1992.

[159] G.Z. Hertz and G.D. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15:563–577, 1999.

[160] N. Heuze, S. Olayat, N. Gutman, M.L. Zani, and Y. Courty. Molecular cloning and expression of an alternative hKLK3 transcript coding for a variant protein of prostate-specific antigen. *Cancer Research*, 59: 2820–2824, 1999.

[161] M. H. Heydari and I. H. Sudborough. On the diameter of the pancake network. *Journal of Algorithms*, 25:67–94, 1997.

[162] D.G. Higgins, J.D. Thompson, and T.J. Gibson. Using CLUSTAL for multiple sequence alignments. *Methods in Enzymology*, 266:383–402, 1996.

[163] D.S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Communication of ACM*, 18:341–343, 1975.

[164] D.S. Hirschberg. Algorithms for the longest common subsequence problem. *Journal of ACM*, 24:664–675, 1977.

[165] J.D. Hoheisel, E. Maier, R. Mott, L. McCarthy, A.V. Grigoriev, L. C. Schalkwyk, D. Nizetic, F. Francis, and H. Lehrach. High resolution cosmid and P1 maps spanning the 14 Mb genome of the fission yeast *S. pombe*. *Cell*, 73:109–120, 1993.

[166] S. Hopper, R.S. Johnson, J.E. Vath, and K. Biemann. Glutaredoxin from rabbit bone marrow. *Journal of Biological Chemistry*, 264:20438–20447, 1989.

[167] Y. Hu, L.R. Tanzer, J. Cao, C.D. Geringer, and R.E. Moore. Use of long RT-PCR to characterize splice variant mRNAs. *Biotechniques*, 25:224–229, 1998.

[168] X. Huang, R.C. Hardison, and W. Miller. A space-efficient algorithm for local similarities. *Computer Applications in Biosciences*, 6:373–381, 1990.

[169] X. Huang and W. Miller. A time-efficient, linear-space local similarity algorithm. *Advances in Applied Mathematics*, 12:337–357, 1991.

[170] T.J. Hubbard, A.M. Lesk, and A. Tramontano. Gathering them into the fold. *Nature Structural Biology*, 4:313, 1996.

[171] E. Hubbell. Multiplex sequencing by hybridization. *Journal of Computational Biology*, 8, 2000.

[172] E. Hubbell and P.A. Pevzner. Fidelity probes for DNA arrays. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 113–117, Heidelberg, Germany, August 1999. AAAI Press.

[173] T.J. Hudson, L.D. Stein, S.S. Gerety, J. Ma, A.B. Castle, J. Silva, D.K. Slonim, R. Baptista, L. Kruglyak, S.H. Xu, X. Hu, A.M.E. Colbert, C. Rosenberg, M.P. Reeve-Daly, S. Rozen, L. Hui, X. Wu, C. Vestergaard, K.M. Wilson, and J.S. Bae et al. An STS-based map of the human genome. *Science*, 270:1945–1954, 1995.

[174] J.W. Hunt and T.G. Szymanski. A fast algorithm for computing longest common subsequences. *Communication of ACM*, 20:350–353, 1977.

[175] R.M. Idury and M.S. Waterman. A new algorithm for DNA sequence assembly . *Journal of Computational Biology*, 2:291–306, 1995.

[176] C. Iseli,C.V. Jongeneel, and P. Bucher. ESTScan: a program for detecting, evaluating and reconstructing potential coding regions in EST sequences. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 138–148, Heidelberg, Germany, August 6-10 1999. AAAI Press.

[177] A.G. Ivanov. Distinguishing an approximate word's inclusion on Turing machine in real time. *Izvestiia Academii Nauk SSSR, Series Math.*, 48:520–568, 1984.

[178] A. Jauch, J. Wienberg, R. Stanyon, N. Arnold, S. Tofanelli, T. Ishida, and T. Cremer. Reconstruction of genomic rearrangements in great apes gibbons by chromosome painting. *Proceedings of the National Academy of Sciences USA*, 89:8611–8615, 1992.

[179] T. Jiang and R.M. Karp. Mapping clones with a given ordering or interleaving. *Algorithmica*, 21:262–284, 1998.

[180] T. Jiang and M. Li. Approximating shortest superstrings with constraints. *Theoretical Computer Science*, 134: 473–491, 1994.

[181] T. Jiang and M. Li. DNA sequencing and string learning. *Mathematical Systems Theory*, 29:387–405, 1996.

[182] R.J. Johnson and K. Biemann. Computer program (SEQPEP) to aid in the interpretation of high-energy collision tandem mass spectra of peptides. *Biomedical and Environmental Mass Spectrometry*, 18:945–957, 1989.

[183] Y.W. Kan and A.M. Dozy. Polymorphism of DNA sequence adjacent to human beta-globin structural gene: relationship to sickle mutation. *Proceedings of the National Academy of Sciences USA*, 75:5631–5635, 1978.

[184] S.K. Kannan and E.W. Myers. An algorithm for locating nonoverlapping regions of maximum alignment score. *SIAM Journal on Computing*, 25:648–662, 1996.

[185] H. Kaplan, R. Shamir, and R.E. Tarjan. Faster and simpler algorithm for sorting signed permutations by reversa In *Proceedings of the Eighth Annual ACM–SIAM Symposium on Discrete Algorithms*, pages 344–351, New Orleans, Louisiana, January 1997.

[186] S. Karlin and S.F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences USA*, 87:2264–2268, 1990.

[187] S. Karlin and G. Ghandour. Multiple-alphabet amino acid sequence comparisons of the immunoglobulin kappa-chain constant domain. *Proceedings of the National Academy of Sciences USA*, 82:8597–8601, 1985.

[188] R.M. Karp, R.E. Miller, and A.L. Rosenberg. Rapid identification of repeated patterns in strings, trees and arrays. In *Proceedings of the Fourth Annual ACM Symposium on Theory of Computing*, pages 125–136, Denver, Colorado, May 1972.

[189] R.M. Karp and M.O. Rabin. Efficient randomized pattern-matching algorithms. *IBM Journal of Research and Development*, 31:249–260, 1987.

[190] R.M. Karp and R. Shamir. Algorithms for optical mapping. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors *Proceedings of the Second Annual International Conference on Computational Molecular Biology (RECOMB-98)*, pages 117–124, New York, New York, March 1998. ACM Press.

[191] J. Kececioglu and R. Ravi. Of mice and men: Evolutionary distances between genomes under translocation. In *Proceedings of the 6th Annual ACM SIAM Symposium on Discrete Algorithms*, pages 604–613, New York, New York, 1995.

[192] J. Kececioglu and D. Sankoff. Exact and approximation algorithms for the reversal distance between two permutations. In *Fourth Annual Symposium on Combinatorial Pattern Matching*, volume 684 of *Lecture Notes in Computer Science*, pages 87–105, Padova, Italy, 1993. Springer-Verlag.

[193] J. Kececioglu and D. Sankoff. Efficient bounds for oriented chromosome inversion distance. In *Fifth Annual Symposium on Combinatorial Pattern Matching*, volume 807 of *Lecture Notes in Computer Science*, pages 307–325, Asilomar, California, 1994. Springer-Verlag.

[194] J. Kececioglu and D. Sankoff. Exact and approximation algorithms for the inversion distance between two permutations. *Algorithmica*, 13:180–210, 1995.

[195] J.D. Kececioglu and E.W. Myers. Combinatorial algorithms for DNA sequence assembly. *Algorithmica*, 13: 7–51, 1995.

[196] T.J. Kelly and H.O. Smith. A restriction enzyme from *Hemophilus influenzae*. II. Base sequence of the recognition site. *Journal of Molecular Biology*, 51:393–409, 1970.

[197] K. Khrapko, Y. Lysov, A. Khorlin, V. Shik, V. Florent'ev, and A. Mirzabekov. An oligonucleotide hybridization approach to DNA sequencing. *FEBS Letters*, 256:118–122, 1989.

[198] J.F.C. Kingman. Subadditive ergodic theory. *Annals of Probability*, 6:883–909, 1973.

[199] J. Kleffe and M. Borodovsky. First and second moment of counts of words in random texts generated by Markov chains. *Computer Applications in Biosciences*, 8:433–441, 1992.

[200] M. Knill, W.J. Bruno, and D.C. Torney. Non-adaptive group testing in the presence of errors. *Discrete Applied Mathematics*, 88:261–290, 1998.

[201] D.E. Knuth. Permutations, matrices and generalized Young tableaux. *Pacific Journal of Mathematics*, 34: 709–727, 1970.

[202] D.E. Knuth. *The Art of Computer Programming*, chapter 2. Addison-Wesley, second edition, 1973.

[203] D.E. Knuth, J.H. Morris, and V.R. Pratt. Fast pattern matching in strings. *SIAM Journal on Computing*, 6: 323–350, 1977.

[204] Y. Kohara, K. Akiyama, and K. Isono. The physical map of the whole E. coli chromosome: application of a new strategy for rapid analysis and sorting of a large genomic library. *Cell*, 50:495–508, 1987.

[205] G.A. Komatsoulis and M.S. Waterman. Chimeric alignment by dynamic programming: Algorithm and biological uses. In *Proceedings of the First Annual International Conference on Computational Molecular Biology (RECOMB-97)*, pages 174–180, Santa Fe, New Mexico, January 1997. ACM Press.

[206] A. Kotzig. Moves without forbidden transitions in a graph. *Matematicky Casopis*, 18:76–80, 1968.

[207] R.G. Krishna and F. Wold. Posttranslational modifications. In R.H. Angeletti, editor, *Proteins - Analysis and Design*, pages 121–206. Academic Press, 1998.

[208] A. Krogh, M. Brown, I.S. Mian, K. Sjölander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology*, 235:1501–1531, 1994.

[209] A. Krogh, I.S. Mian, and D. Haussler. A Hidden Markov Model that finds genes in E. coli DNA. *Nucleic Acids Research*, 22:4768–4778, 1994.

[210] S. Kruglyak. Multistage sequencing by hybridization. *Journal of Computational Biology*, 5:165–171, 1998.

[211] J.B. Kruskal and D. Sankoff. An anthology of algorithms and concepts for sequence comparison. In D. Sankoff and J.B. Kruskal, editors, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, pages 265–310. Addison-Wesley, 1983.

[212] G.M. Landau and J.P. Schmidt. An algorithm for approximate tandem repeats. In *Fourth Annual Symposium on Combinatorial Pattern Matching*, volume 684 of *Lecture Notes in Computer Science*, pages 120–133, Padova, Italy, 2-4 June 1993. Springer-Verlag.

[213] G.M. Landau and U. Vishkin. Efficient string matching in the presence of errors. In *26th Annual Symposium on Foundations of Computer Science*, pages 126–136, Los Angeles, California, October 1985.

[214] E.S. Lander and M.S. Waterman. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics*, 2:231–239, 1988.

[215] K. Lange, M. Boehnke, D.R. Cox, and K.L. Lunetta. Statistical methods for polyploid radiation hybrid mapping. *Genome Research*, 5:136–150, 1995.

[216] E. Lawler and S. Sarkissian. Adaptive error correcting codes based on cooperative play of the game of "Twenty Questions Game with a Liar". In *Proceedings of Data Compression Conference DCC '95*, page 464, Los Alamitos, California, 1995. IEEE Computer Society Press.

[217] C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald, and J.C. Wootton. Detecting subtle sequence signals: a Gibbs saumpling strategy for multiple alignment. *Science*, 262:208–214, October 1993.

[218] J.K. Lee, V. Dancik, and M.S. Waterman. Estimation for restriction sites observed by optical mapping using reversible-jump Markov chain Monte Carlo. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, *Proceedings of the Second Annual International Conference on Computational Molecular Biology (RECOMB-98)*, pages 147–152, New York, New York, March 1998. ACM Press.

[219] V.I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 6:707–710, 1966.

[220] B. Lewin. *Genes VII*. Oxford University Press, 1999.

[221] M. Li, B. Ma, and L. Wang. Finding similar regions in many strings.In *Proceedings of the 31st ACM Annual Symposium on Theory of Computing*, pages 473–482, Atlanta, Georgia, May 1999.

[222] S.Y.R. Li. A martingale approach to the study of ocurrence of sequence patterns in repeated experiments. *Annals of Probability*, 8:1171–1176, 1980.

[223] J. Lingner, T.R. Hughes, A. Shevchenko, M. Mann, V. Lundblad, and T.R.Cech. Reverse transcriptase motifs in the catalytic subunit of telomerase. *Science*, 276:561–567, 1997.

[224] D.J. Lipman, S. F Altschul, and J.D. Kececioglu. A tool for multiple sequence alignment. *Proceedings of the National Academy of Sciences USA*, 86:4412–4415, 1989.

[225] D.J. Lipman and W.R. Pearson. Rapid and sensitive protein similarity searches. *Science*, 227:1435–1441, 1985.

[226] R.J. Lipshutz, D. Morris, M. Chee, E. Hubbell, M.J. Kozal, N. Shah, N. Shen, R. Yang, and S.P.A. Fodor. Using oligonucleotide probe arrays to access genetic diversity. *Biotechniques*, 19:442–447, 1995.

[227] B.F. Logan and L.A. Shepp. A variational problem for random Young tableaux. *Advances in Mathematics*, 26:206–222, 1977.

[228] Y. Lysov, V. Florent'ev, A. Khorlin, K. Khrapko, V. Shik, and A. Mirzabekov. DNA sequencing by hybridization with oligonucleotides. *Doklady Academy Nauk USSR*, 303:1508–1511, 1988.

[229] C.A. Makaroff and J.D. Palmer. Mitochondrial DNA rearrangements and transcriptional alterations in the male sterile cytoplasm of Ogura radish. *Molecular Cellular Biology*, 8:1474–1480, 1988.

[230] M. Mann and M. Wilm. Error-tolerant identification of peptides in sequence databases by peptide sequence tags. *Analytical Chemistry*, 66:4390–4399, 1994.

[231] M. Mann and M. Wilm. Electrospray mass-spectrometry for protein characterization. *Trends in Biochemical Sciences*, 20:219–224, 1995.

[232] D. Margaritis and S.S. Skiena. Reconstructing strings from substrings in rounds. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 613–620, Los Alamitos, California, October 1995.

[233] A.M. Maxam and W. Gilbert. A new method for sequencing DNA. *Proceedings of the National Academy of Sciences USA*, 74:560–564, 1977.

[234] G. Mayraz and R. Shamir. Construction of physical maps from oligonucleotide fingerprints data. *Journal of Computational Biology*, 6:237–252, 1999.

[235] F.R. McMorris, C. Wang, and P. Zhang. On probe interval graphs. *Discrete Applied Mathematics*, 88:315–324, 1998.

[236] W. Miller and E.W. Myers. Sequence comparison with concave weighting functions. *Bulletin of Mathematical Biology*, 50:97–120, 1988.

[237] A. Milosavljevic and J. Jurka. Discovering simple DNA sequences by the algorithmic significance method. *Computer Applications in Biosciences*, 9:407–411, 1993.

[238] B. Mirkin and F.S. Roberts. Consensus functions and patterns in molecular sequences. *Bulletin of Mathematical Biology*, 55:695–713, 1993.

[239] A.A. Mironov and N.N. Alexandrov. Statistical method for rapid homology search. *Nucleic Acids Research*, 16:5169–5174, 1988.

[240] A.A. Mironov, J.W. Fickett, and M.S. Gelfand. Frequent alternative splicing of human genes. *Genome Research*, 9:1288–1293, 1999.

[241] A.A. Mironov and P.A. Pevzner. SST versus EST in gene recognition. *Microbial and Comparative Genomics*, 4:167–172, 1999.

[242] A.A. Mironov, M.A. Roytberg, P.A. Pevzner, and M.S. Gelfand. Performance guarantee gene predictions via spliced alignment. *Genomics*, 51:332–339, 1998.

[243] S. Muthukrishnan and L. Parida. Towards constructing physical maps by optical mapping: An effective, simple, combinatorial approach. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, *Proceedings of the First Annual International Conference on Computational Molecular Biology (RECOMB-97)*, pages 209–219, Santa Fe, New Mexico, January 1997. ACM Press.

[244] M. Muzio, A.M. Chinnaiyan, F.C. Kischkel, K. O'Rourke, A. Shevchenko, J. Ni, C. Scaffidi, J. D. Bretz, M. Zhang, R. Gentz, M. Mann, P.H. Krammer, M. E. Peter, and V. M. Dixit. FLICE, a novel FADD-homologous ICE/CED-3-like protease, is recruited to the CD95 (Fas/APO-1) death-inducing signaling complex. *Cell*, 85: 817–827, 1996.

[245] E.W. Myers. A sublinear algorithm for approximate keyword searching. *Algorithmica*, 12:345–374, 1994.

[246] E.W. Myers and W. Miller. Optimal alignments in linear space. *Computer Applications in Biosciences*, 4:11–17, 1988.

[247] G. Myers. Whole genome shotgun sequencing. *IEEE Computing in Science and Engineering*, 1:33–43, 1999.

[248] J.H. Nadeau and B.A. Taylor. Lengths of chromosomal segments conserved since divergence of man and mouse. *Proceedings of the National Academy of Sciences USA*, 81:814–818, 1984.

[249] K. Nakata, M. Kanehisa, and C. DeLisi. Prediction of splice junctions in mRNA sequences. *Nucleic Acids Research*, 13:5327–5340, 1985.

[250] D. Naor and D. Brutlag. On near-optimal alignments of biological sequences. *Journal of Computational Biology*, 1:349–366, 1994.

[251] S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

[252] L. Newberg and D. Naor. A lower bound on the number of solutions to the probed partial digest problem. *Advances in Applied Mathematics*, 14:172–183, 1993.

[253] R. Nussinov, G. Pieczenik, J.R. Griggs, and D.J. Kleitman. Algorithms for loop matchings. *SIAM Journal on Applied Mathematics*, 35:68–82, 1978.

[254] S. O'Brien and J. Graves. Report of the committee on comparative gene mapping in mammals. *Cytogenetics and Cell Genetics*, 58:1124–1151, 1991.

[255] S. Ohno. *Sex chromosomes and sex-linked genes*. Springer-Verlag, 1967.

[256] S. Ohno, U. Wolf, and N.B. Atkin. Evolution from fish to mammals by gene duplication. *Hereditas*, 59:708–713, 1968.

[257] M.V. Olson, J.E. Dutchik, M.Y. Graham, G.M. Brodeur, C. Helms, M. Frank, M. MacCollin, R.

Scheinman, and T. Frank. Random-clone strategy for genomic restriction mapping in yeast. *Proceedings of the National Academy of Sciences USA*, 83:7826–7830, 1986.

[258] O. Owolabi and D.R. McGregor. Fast approximate string matching. *Software Practice and Experience*, 18: 387–393, 1988.

[259] J.D. Palmer and L.A. Herbon. Plant mitochondrial DNA evolves rapidly in structure, but slowly in sequence. *Journal of Molecular Evolution*, 27:87– 97, 1988.

[260] A.H. Paterson, T.H. Lan, K.P. Reischmann, C. Chang, Y.R. Lin, S.C. Liu, M.D. Burow, S.P. Kowalski, C.S. Katsar, T.A. DelMonte, K.A. Feldmann, K.F. Schertz, and J.F. Wendel. Toward a unified genetic map of higher plants, transcending the monocot-dicot divergence. *Nature Genetics*, 15:380–382, 1996.

[261] S.D. Patterson and R. Aebersold. Mass spectrometric approaches for the identification of gel-separated proteins. *Electrophoresis*, 16:1791–1814, 1995.

[262] H. Peltola, H. Soderlund, and E. Ukkonen. SEQAID: a DNA sequence assembling program based on a mathematical model. *Nucleic Acids Research*, 12:307–321, 1984.

[263] M. Perlin and A. Chakravarti. Efficient construction of high-resolution physical maps from yeast artificial chromosomes using radiation hybrids: inner product mapping. *Genomics*, 18:283–289, 1993.

[264] P.A. Pevzner. *l*-tuple DNA sequencing: computer analysis. *Journal of Biomolecular Structure and Dynamics*, 7:63–73, 1989.

[265] P.A. Pevzner. Multiple alignment, communication cost, and graph matching. *SIAM Journal on Applied Mathematics*, 52:1763–1779, 1992.

[266] P.A. Pevzner. Statistical distance between texts and filtration methods in rapid similarity search algorithm. *Computer Applications in Biosciences*, 8:121–27, 1992.

[267] P.A. Pevzner. DNA physical mapping and alternating Eulerian cycles in colored graphs. *Algorithmica*, 13: 77–105, 1995.

[268] P.A. Pevzner. DNA statistics, overlapping word paradox and Conway equation. In H. A. Lim, J. W. Fickett, C.R. Cantor, and R.J. Robbins, editors, *Proceedings of the Second International Conference on Bioinformatics, Supercomputing, and Complex Genome Analysis*, pages 61–68, St. Petersburg Beach, Florida, June 1993. World Scientific.

[269] P.A. Pevzner, M.Y. Borodovsky, and A.A. Mironov. Linguistics of nucleotide sequences. I: The significance of deviations from mean statistical characteristics and prediction of the frequencies of occurrence of words. *Journal of Biomolecular Structure and Dynamics*, 6:1013–1026, 1989.

[270] P.A. Pevzner, V. Dancik, and C.L. Tang. Mutation-tolerant protein identification by mass-spectrometry. In R. Shamir, S. Miyano, S. Istrail, P. A. Pevzner, and M. S. Waterman, editors, *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology (RECOMB-00)*, pages 231–236, Tokyo, Japan, April 2000. ACM Press.

[271] P.A. Pevzner and R. Lipshutz. Towards DNA sequencing chips. In *Proceedings of the 19th International Conference on Mathematical Foundations of Computer Science*, volume 841 of *Lecture Notes in Computer Science*, pages 143–158, Kosice, Slovakia, 1994.

[272] P.A. Pevzner, Y. Lysov, K. Khrapko, A. Belyavski, V. Florentiev, and A. Mirzabekov. Improved chips for sequencing by hybridization. *Journal of Biomolecular Structure and Dynamics*, 9:399–410, 1991.

[273] P.A. Pevzner and M.S. Waterman. Generalized sequence alignment and duality. *Advances in Applied Mathematics*, 14(2):139–171, 1993.

[274] P.A. Pevzner and M.S. Waterman. Multiple filtration and approximate pattern matching. *Algorithmica*, 13: 135–154, 1995.

[275] P.A. Pevzner and M.S. Waterman. Open combinatorial problems in computational molecular biology. In *Third Israeli Symposium on the Theory of Computing and Systems*, Tel-Aviv, Israel, January 1995.

[276] S. Pilpel. Descending subsequences of random permutations. *Journal of Combinatorial Theory, Series A*, 53:96–116, 1990.

[277] A. Pnueli, A. Lempel, and S. Even. Transitive orientation of graphs and identification of permutation graphs. *Canadian Journal of Mathematics*, 23:160–175, 1971.

[278] J.H. Postlethwait, Y.L. Yan, M.A. Gates, S. Horne, A. Amores, A. Brownlie, A. Donovan, E.S. Egan, A. Force, Z. Gong, C. Goutel, A. Fritz, R. Kelsh, E. Knapik, E. Liao, B. Paw, D. Ransom, A. Singer, M. Thomson, T.S. Abduljabbar, P. Yelick, D. Beier, J.S. Joly, D. Larhammar, and F. Rosa et al. Vertebrate genome evolution and the zebrafish gene map. *Nature Genetics*, 345-349:18, 1998.

[279] A. Poustka, T. Pohl, D.P. Barlow, G. Zehetner, A. Craig, F. Michiels, E. Ehrich, A.M. Frischauf, and H. Lehrach. Molecular approaches to mammalian genetics. *Cold Spring Harbor Symposium on Quantitative Biology*, 51:131–139, 1986.

[280] F.P. Preparata, A.M. Frieze, and E. Upfal. On the power of universal bases in sequencing by hybridization. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, *Proceedings of the Third Annual International Conference on Computational Molecular Biology (RECOMB-99)*, pages 295–301, Lyon, France, April 1999. ACM Press.

[281] B. Prum, F. Rudolphe, and E. De Turckheim. Finding words with unexpected frequences in DNA sequences. *Journal of Royal Statistical Society, Series B*, 57:205–220, 1995.

[282] M. Regnier and W. Szpankowski. On the approximate pattern occurrences in a text. In *Compression and Complexity of Sequences 1997*, pages 253–264, 1998.

[283] K. Reinert, H.-P. Lenhof, P. Mutzel, K. Mehlhorn, and J.D. Kececioglu. A branch-and-cut algorithm for multiple sequence alignment. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, *Proceedings of the First Annual International Conference on Computational Molecular Biology (RECOMB-97)*, pages 241–250, Santa Fe, New Mexico, January 1997. ACM Press.

[284] G. Rettenberger, C. Klett, U. Zechner, J. Kunz, W. Vogel, and H. Hameister. Visualization of the conservation of synteny between humans and pigs by hetereologous chromosomal painting. *Genomics*, 26:372–378, 1995.

[285] I. Rigoutsos and A. Floratos. Combinatorial pattern discovery in biological sequences. *Bioinformatics*, 14:55–67, 1998.

[286] J.C. Roach, C. Boysen, K. Wang, and L. Hood. Pairwise end sequencing: a unified approach to genomic mapping and sequencing. *Genomics*, 26:345–353, 1995

[287] G. de E. Robinson. On representations of the symmetric group. *American Journal of Mathematics*, 60:745–760, 1938.

[288] E. Rocke and M. Tompa. An algorithm for finding novel gapped motifs in DNA sequences. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, *Proceedings of the Second Annual International Conference on Computational Molecular Biology (RECOMB-98)*, pages 228–233, New York, New York, March 1998. ACM Press.

[289] J. Rosenblatt and P.D. Seymour. The structure of homometric sets. *SIAM Journal on Alg. Discrete Methods*, 3:343–350, 1982.

[290] M.A. Roytberg. A search for common pattern in many sequences. *Computer Applications in Biosciences*, 8:57–64, 1992.

[291] A.R. Rubinov and M.S. Gelfand. Reconstruction of a string from substring precedence data. *Journal of Computational Biology*, 2:371–382, 1995.

[292] B.E. Sagan. *The Symmetric Group: Representations, Combinatorial Algorithms, and Symmetric Functions*. Wadsworth Brooks Cole Mathematics Series, 1991.

[293] M.F. Sagot, A. Viari, and H. Soldano. Multiple sequence comparison—a peptide matching approach. *Theoretical Computer Science*, 180:115–137, 1997.

[294] T. Sakurai, T. Matsuo, H. Matsuda, and I. Katakuse. PAAS 3: A computer program to determine probable sequence of peptides from mass spectrometric data. *Biomedical Mass Spectrometry*, 11:396–399, 1984.

[295] S.L. Salzberg, A.L. Delcher, S. Kasif, and O. White. Microbial gene identification using interpolated Markov models. *Nucleic Acids Research*, 26:544–548, 1998.

[296] S.L. Salzberg, D.B. Searls, and S. Kasif. *Computational Methods in Molecular Biology*. Elsevier, 1998.

[297] F. Sanger, S. Nilken, and A.R. Coulson. DNA sequencing with chain terminating inhibitors. *Proceedings of the National Academy of Sciences USA*, 74:5463–5468, 1977.

[298] D. Sankoff. Minimum mutation tree of sequences. *SIAM Journal on Applied Mathematics*, 28:35–42, 1975.

[299] D. Sankoff. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM Journal on Applied Mathematics*, 45:810–825, 1985.

[300] D. Sankoff. Edit distance for genome comparison based on non-local operations. In *Third Annual Symposium on Combinatorial Pattern Matching*, volume 644 of *Lecture Notes in Computer Science*, pages 121–135, Tucson, Arizona, 1992. Springer-Verlag.

[301] D. Sankoff and M. Blanchette. Multiple genome rearrangements. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, *Proceedings of the Second Annual International Conference on Computational Molecular Biology (RECOMB-98)*, pages 243–247, New York, New York, March 1998. ACM Press.

[302] D. Sankoff, R. Cedergren, and Y. Abel. Genomic divergence through gene rearrangement. In *Molecular Evolution: Computer Analysis of Protein and Nucleic Acid Sequences*, chapter 26, pages 428–438. Academic Press, 1990.

[303] D. Sankoff and M. Goldstein. Probabilistic models of genome shuffling. *Bulletin of Mathematical Biology*, 51:117–124, 1989.

[304] D. Sankoff, G. Leduc, N. Antoine, B. Paquin, B. Lang, and R. Cedergren. Gene order comparisons for phylogenetic inference: Evolution of the mitochondrial genome. *Proceedings of the National Academy of Sciences USA*, 89:6575–6579, 1992.

[305] D.Sankoff and S.Mainville.Common subsequences and monotone subse quences.In D.Sankoff and J.B.Kruskal, editors, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, pages 363–365. Addison-Wesley, 1983.

[306] C. Schensted. Longest increasing and decreasing subsequences. *Canadian Journal of Mathematics*, 13:179–191, 1961.

[307] H. Scherthan, T. Cremer, U. Arnason, H. Weier, A. Lima de Faria, and L. Fronicke. Comparative chromosomal painting discloses homologous segments in distantly related mammals. *Nature Genetics*, 6:342–347, 1994.

[308] J.P. Schmidt. All highest scoring paths in weighted grid graphs and their application to finding all approximate repeats in strings. *SIAM Journal on Computing*, 27:972–992, 1998.

[309] W. Schmitt and M.S. Waterman. Multiple solutions of DNA restriction mapping problem. *Advances in Applid Mathematics*, 12:412–427, 1991.

[310] M. Schoniger and M.S. Waterman. A local algorithm for DNA sequence alignment with inversions. *Bulletin of Mathematical Biology*, 54:521–536, 1992.

[311] D.C. Schwartz, X. Li, L.I. Hernandez, S.P. Ramnarain, E.J. Huff, and Y.K. Wang. Ordered restriction maps of Saccharomyces cerevisiae chromosomes constructed by optical mapping. *Science*, 262:110–114, 1993.

[312] D. Searls and S. Dong. A syntactic pattern recognition system for DNA sequences. In H.A. Lim, J. W. Fickett, C.R. Cantor, and R.J. Robbins, editors, *Proceedings of the Second International Conference on Bioinformatics, Supercomputing, and Complex Genome Analysis*, pages 89–102, St. Petersburg Beach, Florida, June 1993. World Scientific.

[313] D. Searls and K. Murphy. Automata-theoretic models of mutation and alignment. In *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pages 341–349, Cambridge, England, 1995.

[314] S.S. Skiena, W.D. Smith, and P. Lemke. Reconstructing sets from interpoint distances. In *Proceedings of Sixth Annual Symposium on Computational Geometry*, pages 332–339, Berkeley, California, June, 1990.

[315] S.S. Skiena and G. Sundaram. A partial digest approach to restriction site mapping. *Bulletin of Mathematical Biology*, 56:275–294, 1994.

[316] S.S. Skiena and G. Sundram. Reconstructing strings from substrings. *Journal of Computational Biology*, 2:333–354, 1995.

[317] D. Slonim, L. Kruglyak, L. Stein, and E. Lander. Building human genome maps with radiation hybrids. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, *Proceedings of the First Annual International Conference on Computational Molecular Biology (RECOMB-97)*, pages 277–286, Santa Fe, New Mexico, January 1997. ACM Press.

[318] H.O. Smith, T.M. Annau, and S. Chandrasegaran. Finding sequence motifs in groups of functionally related proteins. *Proceedings of the National Academy of Sciences USA*, 87:826–830, 1990.

[319] H.O. Smith and K.W. Wilcox. A restriction enzyme from Hemophilus influenzae. I. Purification and general properties. *Journal of Molecular Biology*, 51:379–391, 1970.

[320] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[321] E.E. Snyder and G.D. Stormo. Identification of coding regions in genomic DNA sequences: an application of dynamic programming and neural networks. *Nucleic Acids Research*, 21:607–613, 1993.

[322] E.E. Snyder and G.D. Stormo. Identification of protein coding regions in genomic DNA. *Journal of Molecular Biology*, 248:1–18, 1995.

[323] V.V. Solovyev, A.A. Salamov, and C.B. Lawrence. Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. *Nucleic Acids Research*, 22:5156–63, 1994.

[324] E.L. Sonnhammer, S.R. Eddy, and R. Durbin. Pfam: a comprehensive database of protein domain families based on seed alignments. *Proteins*, 28:405–420, 1997.

[325] E. Southern. United Kingdom patent application GB8810400. 1988.

[326] R. Staden. Methods for discovering novel motifs in nucleic acid seqences. *Computer Applications in Biosciences*, 5:293–298, 1989.

[327] R. Staden and A.D. McLachlan. Codon preference and its use in identifying protein coding regions in long DNA sequences. *Nucleic Acids Research*, 10:141–156, 1982.

[328] J.M. Steele. An Efron-Stein inequality for nonsymmetric statistics. *Annals of Statistics*, 14:753–758, 1986.

[329] M. Stefik. Inferring DNA structure from segmentation data. *Artificial Intelligence*, 11:85–144, 1978.

[330] E.E. Stuckle, C. Emmrich, U. Grob, and P.J. Nielsen. Statistical analysis of nucleotide sequences. *Nucleic Acids Research*, 18:6641–6647, 1990.

[331] A.H. Sturtevant and T. Dobzhansky. Inversions in the third chromosome of wild races of *Drosophila pseudoobscura*, and their use in the study of the history of the species. *Proceedings of the National Academy of Sciences USA*, 22:448–450, 1936.

[332] S.H. Sze and P.A. Pevzner. Las Vegas algorithms for gene recognition: subotimal and error tolerant spliced alignment. *Journal of Computational Biology*, 4:297–310, 1997.

[333] J. Tarhio and E. Ukkonen. A greedy approximation algorithm for constructing shortest common superstrings. *Theoretical Computer Science*, 57:131–145, 1988.

[334] J. Tarhio and E. Ukkonen. Boyer-Moore approach to approximate string matching. In J.R. Gilbert and R. Karlsson, editors, *Proceedings of the Second Scandinavian Workshop on Algorithm Theory*, number 447 in Lecture Notes in Computer Science, pages 348–359, Bergen, Norway, 1990. Springer-Verlag.

[335] J.A. Taylor and R.S. Johnson. Sequence database searches via *de novo* peptide sequencing by tandem mass spectrometry. *Rapid Communications in Mass Spectrometry*, 11:1067–1075, 1997.

[336] W.R. Taylor. Multiple sequence alignment by a pairwise algorithm. *Computer Applications in Biosciences*, 3:81–87, 1987.

[337] S.M. Tilghman, D.C. Tiemeier, J.G. Seidman, B.M. Peterlin, M. Sullivan, J.V. Maizel, and P. Leder. Intervening sequence of DNA identified in the structural portion of a mouse beta-globin gene. *Proceedings of the National Academy of Sciences USA*, 75:725–729, 1978.

[338] M. Tompa. An exact method for finding short motifs in sequences with application to the Ribosome Binding Site problem. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 262–271, Heidelberg, Germany, August 1999. AAAI Press.

[339] E. Uberbacher and R. Mural. Locating protein coding regions in human DNA sequences by a multiple sensor-neural network approach. *Proceedings of the National Academy of Sciences USA*, 88:11261–11265, 1991.

[340] E. Ukkonen. Approximate string matching with q-grams and maximal matches. *Theoretical Computer Science*, 92:191–211, 1992.

[341] S. Ulam. Monte-Carlo calculations in problems of mathematical physics.In *Modern mathematics for the engineer*, pages 261–281. McGraw-Hill, 1961.

[342] A.M. Vershik and S.V. Kerov. Asymptotics of the Plancherel measure of the symmetric group and the limiting form of Young tableaux. *Soviet Mathematical Doklady*, 18:527–531, 1977.

[343] M. Vihinen. An algorithm for simultaneous comparison of several sequences. *Computer Applications in Biosciences*, 4:89–92, 1988.

[344] M. Vingron and P. Argos. Motif recognition and alignment for many sequences by comparison of dot-matrices. *Journal of Molecular Biology*, 218:33–43, 1991.

[345] M. Vingron and P.A. Pevzner. Multiple sequence comparison and consistency on multipartite graphs. *Advances in Applied Mathematics*, 16:1–22, 1995.

[346] M. Vingron and M.S. Waterman. Sequence alignment and penalty choice. Review of concepts, studies and implications. *Journal of Molecular Biology*, 235:1–12, 1994.

[347] T.K. Vintsyuk. Speech discrimination by dynamic programming. *Comput.*, 4:52–57, 1968.

[348] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, 1967.

[349] D.G. Wang, J.B. Fan, C.J. Siao, A. Berno, P. Young, R. Sapolsky, G. Ghandour, N. Perkins, E. Winchester, J. Spencer, L. Kruglyak, L. Stein, L. Hsie, T. Topaloglou, E. Hubbell, E. Robinson, M. Mittmann, M.S. Morris, N. Shen, D. Kilburn, J. Rioux, C. Nusbaum, S. Rozen, T.J. Hudson, and E.S. Lander et al. Large-scale identification, mapping, and genotyping of single-nucleotide polymorphisms in the human genome. *Science*, 280:1074–1082, 1998.

[350] L. Wang and D. Gusfield. Improved approximation algorithms for tree alignment. In *Seventh Annual Symposium on Combinatorial Pattern Matching*, volume 1075 of *Lecture Notes in Computer Science*, pages 220–233, Laguna Beach, California, 10-12 June 1996. Springer-Verlag.

[351] L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1:337–348, 1994.

[352] L. Wang, T. Jiang, and E.L. Lawler. Approximation algorithms for tree alignment with a given phylogeny. *Algorithmica*, 16:302–315, 1996.

[353] M.D. Waterfield, G.T. Scrace, N. Whittle, P. Stroobant, A. Johnsson, A. Wasteson, B. Westermark, C.H. Heldin, J.S. Huang, and T.F. Deuel. Platelet-derived growth factor is structurally related to the putative transforming protein p28sis of simian sarcoma virus. *Nature*, 304:35–39, 1983.

[354] M.S. Waterman. Secondary structure of single-stranded nucleic acids. *Studies in Foundations and Combinatorics, Advances in Mathematics Supplementary Studies*, 1:167–212, 1978.

[355] M.S. Waterman. Sequence alignments in the neighborhood of the optimum with general application to dynamic programming. *Proceedings of the National Academy of Sciences USA*, 80:3123–3124, 1983.

[356] M.S. Waterman. Efficient sequence alignment algorithms. *Journal of Theoretical Biology*, 108:333–337, 1984.

[357] M.S. Waterman. *Introduction to Computational Biology*. Chapman Hall, 1995.

[358] M.S. Waterman, R. Arratia, and D. J. Galas. Pattern recognition in several sequences: consensus and alignment. *Bulletin of Mathematical Biology*, 46:515–527, 1984.

[359] M.S. Waterman and M. Eggert. A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. *Journal of Molecular Biology*, 197:723–728, 1987.

[360] M.S. Waterman, M. Eggert, and E. Lander. Parametric sequence comparisons. *Proceedings of the National Academy of Sciences USA*, 89:6090–6093, 1992.

[361] M.S. Waterman and J.R. Griggs. Interval graphs and maps of DNA. *Bulletin of Mathematical Biology*, 48:189–195, 1986.

[362] M.S. Waterman and M.D. Perlwitz. Line geometries for sequence comparisons. *Bulletin of Mathematical Biology*, 46:567–577, 1984.

[363] M.S. Waterman and T.F. Smith. Rapid dynamic programming algorithms for RNA secondary structure. *Advances in Applied Mathematics*, 7:455–464, 1986.

[364] M.S. Waterman, T.F. Smith, and W.A. Beyer. Some biological sequence metrics. *Advances in Mathematics*, 20:367–387, 1976.

[365] M.S. Waterman and M. Vingron. Rapid and accurate estimates of statistical significance for sequence data base searches. *Proceedings of the National Academy of Sciences USA*, 91:4625–4628, 1994.

[366] G.A. Watterson, W.J. Ewens, T.E. Hall, and A. Morgan. The chromosome inversion problem. *Journal of Theoretical Biology*, 99:1-7, 1982.

[367] J. Weber and G. Myers. Whole genome shotgun sequencing. *Genome Research*, 7:401–409, 1997.

[368] W.J. Wilbur and D.J. Lipman. Rapid similarity searches of nucleic acid protein data banks. *Proceedings of the National Academy of Sciences USA*, 80:726–730, 1983.

[369] K.H. Wolfe and D.C. Shields. Molecular evidence for an ancient duplication of the entire yeast genome. *Nature*, 387:708–713, 1997.

[370] F. Wolfertstetter, K. Frech, G. Herrmann, and T. Werner. Identification of functional elements in unaligned nucleic acid sequences. Computer Applications in Biosciences, 12:71–80, 1996.

[371] S. Wu and U. Manber. Fast text searching allowing errors. Communication of ACM, 35:83–91, 1992.

[372] G. Xu, S.H. Sze, C.P. Liu, P.A. Pevzner, and N. Arnheim. Gene hunting without sequencing genomic clones: finding exon boundaries in cDNAs. Genomics, 47:171–179, 1998.

[373] J. Yates, J. Eng, and A. McCormack. Mining genomes: Correlating tandem mass-spectra of modified and unmodified peptides to sequences in nucleotide databases. Analytical Chemistry, 67:3202–3210, 1995.

[374] J. Yates, J. Eng, A. McCormack, and D. Schieltz. Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database. Analytical Chemistry, 67:1426–1436, 1995.

[375] J. Yates, P. Griffin, L. Hood, and J. Zhou. Computer aided interpretation of low energy MS/MS mass spectra of peptides. In J.J. Villafranca, editor, Techniques in Protein Chemistry II, pages 477–485. Academic Press, 1991.

[376] P. Zhang, E.A. Schon, S.G. Fischer, E. Cayanis, J. Weiss, S. Kistler, and P. E. Bourne. An algorithm based on graph theory for the assembly of contigs in physical mapping. Computer Applications in Biosciences, 10:309–317, 1994.

[377] Z. Zhang. An exponential example for a partial digest mapping algorithm. Journal of Computational Biology, 1:235–239, 1994.

[378] D. Zidarov, P. Thibault, M.J. Evans, and M.J. Bertrand. Determination of the primary structure of peptides using fast atom bombardment mass spectrometry. Biomedical and Environmental Mass Spectrometry, 19:13–16, 1990.

[379] R. Zimmer and T. Lengauer. Fast and numerically stable parametric alignment of biosequences. In S. Istrail, P.A. Pevzner, and M.S. Waterman, editors, Proceedings of the First Annual International Conference on Computational Molecular Biology (RECOMB-97), pages 344–353, Santa Fe, New Mexico, January 1997. ACM Press.

[380] M. Zuker. RNA folding. Methods in Enzymology, 180:262–288, 1989.

[381] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. Bulletin of Mathematical Biology, 46:591–621, 1984.

索　　引

（以中文汉语拼音排序）

安全反序　safe reversal　153,154,156,158,159
氨基酸　amino acid　1,10,13,14,44
半结点　semi-knot　171-174
半平衡图　semi-balanced graph　25,54
半序集　partially crdered set　84,85
保真探针　fidelity probes　70
堡垒　fortress　160-165
堡垒-结点　fortress-of-knots　166
编辑距离　edit distance　8,10,71,74,102
编辑图　edit graph　75-78,93
标准化局部联配　normalized local alignment　200
标准杨氏表　standard Young tableau　80,201
表现型　phenotype　1-3
不冲突的区间集合　conflict-free interval set　195
不完整表　partial tableau　80,81
不严格重复问题　inexact repeat problem 200
布局　placement　33
部分酶切问题　PDP　6,15,25,26,28
部分酶切问题　Partial Digest Problem　6,15,25,26,28
部分肽　partial peptide　14
参数的序列联配　parametric alignment　92
测序的结束阶段　finishing phase of sequencing　48
插缺　indel　75,77,92
插入　insertion　4,8,21,30,32
查询匹配问题　query matching problem　88
长散布核元件重复　LINE repeat　47
超结点　superknot　166
超序列　supersequence　96,200,202
超障碍　superhurdle　157,158,160-162
成对和得分　Sum-of-Pairs score (SP-score)　97,98
持续堆积杂交　continuous stacking hybridization　57
重叠的词矛盾　overlapping words paradox
重叠连续群　contig　46,47
重复（DNA 中）　repeat (in DNA)　4,21,36,40,45
重排方案　rearrangement scenario　11,13,132,133,135
重新测序　re-sequencing　10,50
重组　recombination　2-4,32
串联质谱技术　MS/MS　179
串联重复　tandem duplication　93,200
串联重复问题　tandem repeat problem　200
磁带翻转问题　pancake flipping problem 135
次优序列联配　suboptimal sequence alignment　93
错配　mismatch　72,75,91,92,96
大尺度的固定化聚合物合成技术　VLSIPS　66
带缺口 l 元组　gapped l-tuple　91
带缺口的信号　gapped signals　116
单完全酶切　Single Complete Digest (SCD)　39
单元素　singleton　138,139
蛋白（水解）酶　protease　44

蛋白质　protein　9,10,13,44,45
蛋白质测序　protein sequencing　13,44,45,207
等价变换　equivalent transformations　124,149,176
递减子序列　decreasing subsequence　200,201
递增序列　increasing subsequence　80
点阵　dot-matrix　89,96,100-102
定位 SBH　positional SBH (PSBH)　126,128,142
定位克隆　positional cloning　126
定位欧拉路问题　positional Eulerian Path Problem　62
定向边（断点图）　oriented edge (breakpoint graph)　141
定向分支　oriented component　147,148,153-155
动态规划　dynamic programming　46,71-74
断点　breakpoint　135-138,141
断点图　breakpoint graph　135-137,141,142
对称多项式　symmetric polynomial　28
对换距离　transposition distance　205
对偶性　duality　84
多基因组的距离问题　Multiple Genomic Distance Problem　176
多面体方法　polyhedral approach　88
多探针　multiprobe　63-66,69
多探针阵列　multiprobe array　65,70
多重分岔　multifork　24
多重联配的得分　score of multiple alignment　96,97
多重酶切问题　Multiple Digest Problem　194
二元阵列　binary array　63,64,198
发出概率　emission probability　112,114,115
翻译　translation　120,177,209
翻译后的修饰　post-translational modifications　177
翻转向量　flip vector　165
反对称路径　anti-symmetric path　185,191
反链　antichain　85,86,88
反谱　reversed spectrum　186,187,191
反序　reversal　132,134-137
反序距离　reversal distance　135-137,141,142
反序排序法　sorting by reversals　135
反序直径　reversal diameter　135,142,145,204
仿射缺口罚分　affine gap penalties　77,115,199
放射杂交图　Radiation Hybrid Mapping　42
分岔图　fork graph　23-25
分而治之　divide-and-conquer　77
分枝概率　branching probability　65,66
符号排列　signed permutations　137-139,145,146
覆盖　cover　5,6,31-34
覆盖数　coverage　41
覆盖图　cover graph　156-158,161
公共反向森林　commom inverted forests　85
公共反向子序列　common inverted subsequences　85
公共森林　common forests　85
供体　donor site　120,122,125,130
共轭半序　conjugate partial orders　85,86,88
共尾部基因组　co-tailed genomes　164-167
共有波峰的个数　shared peaks count　178
共有词分析　consensus word analysis　111
共有区的距离　Distance from Consensus　97
共有字符串问题　Consensus String Problem　111
光导合成　light-directed array synthesis　67
光学作图　optical mapping　28,29
广义排列　generalized permutation　150,152
广义序列联配　generalized sequence alignment　84,85

过滤效率 filtration efficiency 89,90
核苷酸 nucleotide 1,3,4,7,8
盒式反射 cassette reflection 17,20,24
盒式交换 cassette exchange 17,19
候选基因文库 candidate gene library 126
回溯 backtracking 74,78,87
基因 gene 1-5
基因组 genome 1-6
基因组比较 genome comparison 132-134
基因组距离 genomic distance 140,171,173,175,176
基因组排序 genomic sorting 140,141,164
基因组重复 genome duplication 175
基因组重排 genome rearrangement 11
吉布斯抽样 Gibbs sampling 116
加权平均（汉明）距离 TSP Hamming Distance TSP 33
加探针部分酶切作图 Probed Partial Digest Mapping 29
剪接 splicing 10,103,118,119,120
剪接联配 spliced alignment 10,121,123-125
剪接阴影 splicing shadow 127
简单排列 simple permutation 149,152,153,155,163
简化二元阵列 reduced binary array 198
交叉边 interleaving edges 205,206
交叉圈 interleaving cycles 147,155,171
交叉图 interleaving graph 55,146,147,153,155
交错圈 alternating cycle 20,23,26,137,142
交替阵列 alternating array 64
酵母人工染色体 YAC 32
结点 knot 20,166,171-173
解码问题 Decoding Problem 113,203
近似字符串匹配 approximate string matching 88,111
局部联配 local alignment 72,76,92,93,199
矩阵点积 matrix dot product 98
聚合酶链式反应 PCR 209
决策树 decision tree 128
均衡划分 balanced partitioning 199
均匀阵列 uniform array 63,65,69,198
开读框 Open Reading Frame (ORF) 119
可比图 comparability graph 38
可变剪接 alternative splicing 125,128-130
可迁定向 transitive orientation 38
克隆 cloning 4,5,6,13,30
克隆池 pooling 41,42
克隆的指纹图谱 fingerprint of clone 1,39
克隆库 clone library 4,5
克隆异常 clone abnormalities 32
克隆载体 cloning vector 30,209,210
跨越引物 spanning prmer 130
垃圾 DNA junk DNA 9,118
离子类型 ion-type 178,180-184
理论谱 theoretical spectrum 14,177,178,180,186
联配 alignment 10,71,72,75,76
联配得分 alignment score 72,76,78,92,93
链 chain 10,31,33,37,45
裂变 fission 140,141,164,168,175
邻接 adjacency 55,69,101,102,136
临界路 critical path 193
六聚体计数 hexamer count 120
路径覆盖 path cover 39,40,88
旅行（流动）销售员问题 Traveling Salesman Problem (TSP) 33
密码子 codon 120,125,130,208
密码子选择 codon usage 120
嘧啶 pyrimidines 63
模式驱动方法 pattern-driven approach 105
魔力词问题 magic word problem 103,117
母函数 generating function 27
囊肿性纤维化 cystic fibrosis 1,4,8,9,10
黏粒 cosmid 32,195
黏粒重叠群作图 cosmid contig mapping 195

内部反序 internal reversal 164,167
内含子 intron 118,119,120,121,125
内积作图 Inner Product Mapping 196
凝胶电泳 gel-electrophoresis 3,17,50,192,210
欧拉圈 Eulerian cycle 16,20-23
欧拉图 Eulerian graph 54,55,59,60,62
配对序列 mates 17
匹配 match 8,10,14,46,50
片段装配问题 fragment assembly problem 45,197
偏频函数 offset frequency function 182,183
嘌呤 purine 63,69
频繁字符串问题 Frequent String Problem 111
平衡顶点 balanced vertex 20,54
平衡图 balanced graph 20,25,130,142,193
剖面 profile 114-116,119,120
剖面隐马氏模型联配 profile HMM alignment 114
谱的卷积 spectral convolution 186,187
谱积 spectral product 188,191
谱联配 spectral alignment 188-191
谱图 spectrum graph 13,180-183
启动子 promoter 103,104,110,120,208
起始密码子 Start codon 120,130
恰当反序 proper reversal 146,148
前缀反序排序 sorting by prefix reversals 135
前缀反序直径 prefix reversal diameter 135,204
嵌合克隆 chimeric clone 32,37
嵌合联配问题 chimeric alignment problem 200
嵌入 embedding 45,205,206
嵌入中的交叉边 crossing edges in embedding 205
嵌套链杂交 nested strand hybridization 198
强同效集 strongly homometric sets 15,194
区间图 interval graph 31,32,37-39
圈分解 cycle decomposition 136,137,141-143
全局联配 Global alignment 72,76,199-201
缺口 gap 41,47,63,66,69
缺口罚分 gap penalty 77,92,115,124,199
缺口阵列 gapped array 63,66,69
缺失 deletion 10,55,75,77,92
染色体 chromosome 1-5
染色体步查 chromosome walking 4,5
染色体的尾部 tails of chromosome 164
染色体翻转 flipping of chromosome 164
染色体加帽 capping of chromosomes 168
染色体间的边 interchromosomal edge 166
染色体内的边 intrachromosomal edge 166
染色体着色 chromosome painting 141
人类基因组计划 Human Genome Project 44,45,49
融合 fusion 42,140,141,164,168
三角剖分图 triangulated graph 37-39
筛选问题 winnowing problem 93
受体 acceptor site 47,120,128,130
双表 bitableau 80,81
双管鸟枪测序 double-barreled sequencing 46,47
双链 DNA double-stranded DNA 210
双酶切问题 Double Digest Problem 16-18
双酶切问题 DDP 16-18,194
双向间隔图 bipartite interval graph 193
双重过滤 double filtration 91
顺序反射 order reflection 21-23,59
顺序交换 order exchange 21-23,60
肽 peptide 13,14,50,177,178
肽测序问题 Peptide Sequencing Problem 13,14,177,178,180
肽鉴定问题 Peptide Identification Problem 186

肽吟阵列 tiling array 50
肽谱 spectrum of peptide 177
肽序列标签 Peptide Sequence Tag 178
探针 probe 3-7
探针区间图 probe interval graph 195,196
填充 padding 150-152,161,163
通用碱基 universal bases 69
同效集 homometric sets 15,26,27,194
统计距离 statistical distance 93,94
图表调节 diagram adjustment 194
图像重构 image reconstruction 100,101
图一致性问题 Graph Consistency Problem 101
外显子 exon 9,118-121
完全图 complete graph 38,98
网络联配 network alignment 124
惟一探针作图 mapping with nonunique probes 41
无单元素排列 singleton-free permutation 139
无定向边 unoriented edge 166
无定向分支 unoriented component 147,148,154-156
物理图谱 physical map 4,5,12,15,17
细菌人工染色体 Bacterial Artificial Chromosome 32
限制酶 restriction enzyme 3,4,6,15,16
限制酶切位点 restriction site 3,15,104
限制片段 restriction fragments 3,4,17,29,31
限制性酶切图谱 restriction map 6,28,29
限制性片段长度多态性 RFLP 3
限制性片段长度多态性 restriction fragment length polymorphism 3
相变曲线 phase transition curve 92,202
相关多项式 correlation polynomial 105-107,109
相邻1性质 consecutive onse property 32
相似得分 similarity score 74
镶嵌效应 mosaic effect 125
向后算法 backward algorithm 113
向前算法 forward algorithm 113
星联配 star-alignment 97-99
行插人 row insertion 80
序列标签位置 STS 31
序列标签位置 Sequence Tag Site 111
杨氏表 Young tableau 71
杨氏图表 Young diagram 80
一致边 consistent edge 101
一致图 consistent graph 101
遗传标记 genetic markers 3
遗传密码 genetic code 208,209
遗传作图 genetic mapping 1,2,4,43,126
易位 translocation 140,164,165,167,168
隐马氏模型 HMM 43,112-115
隐马氏模型 Hidden Markov Model 43,112-115
隐马氏模型参数估计 parameter estimation for HMM 114
有界阵列 bounded array 198
有效反序 valid reversal 164,206
原核生物 prokaryotes 118,125,208
原子区间 atomic interval 34,35
圆-弧图 circular-arc graph 195
杂交 hybridization 3-7
杂交测序 Sequencing by Hybridization 7,49,51,197
杂交测序 SBH 7,49,51,197
杂交筛选矩阵 hybrid screening matrix 42,43,196
杂交指纹 hybridization fingerprint 33,41
障碍 hurdle 5,33,50,137,146
真核生物 eukaryotes 118,200,208
真图 proper gaph 185
终止密码子 Stop codon 120,130,208
欧拉转换 Euler switch 60,61,62

转录　transcription　44,103,208
状态转移概率矩阵　state transition probability　112
自适应 SBH　adaptive SBH　70,127
自相关多项式　autocorrelation polynomial　105
字符串统计问题　String Statistics Problem　11,111
字符串先后次序数据　strings precedence data　198
组测试问题　Group Testing Problem　42
最长递增子序列　Longest Increasing Subsequence　79,82,83,84
最长共同子序列　Longest Common Subsequence　8,71,73,75,82
最长路径问题　longest path problem　76,185,195
最大片段对　maximal segment pair　90
最短超字符串　Shortest Superstring Problem　7,8,47,62,196
最短覆盖字符串问题　Shortest Covering String Problem　5,31,34,35,40
最短共同超序列　shortest common supersequence　96
最小覆盖　minimum cover　46,85,86
最小熵得分　minimal entropy score　97
2 进 2 出图　2-in-2-out graph　61
2-路　2-path　60-62,96
2-路欧拉集　Euler set of 2-paths　60
2-最优欧拉圈　2-optimal Eulerian cycle　60,61
Alu 重复　Alu repeat　47
Arratia-Steele 猜想　Arratia-Steele conjecture　83,93,201
Baum-Welch 算法　Baum-Welch algorithm　114,116
BEST 定理　BEST theorem　55,93
Catalan 数　Catalan number　57,200
cDNA（互补 DNA）　cDNA　209
CG 岛　CG-island　11,111,112
DNA 片段的谱　spectrum of DNA fragment　52
DNA 阵列　DNA array　7,10,49-51
DNA 阵列的分辨率　resolving power of DNA array　64
DNA 阵列的内存　memory of DNA array　63
Gollan 排列　Gollan permutation　112
Gray 编码　Gray code　67-69
Hamilton 路　Hamiltonian path　50,52,53
l-星　*l*-star　99,100
PCR 引物　PCR primer　126,127,192,209
RNA 折叠　RNA folding　94,202
Sankoff-Mainville 猜想　Sankoff-Mainville conjecture　201
Viterbi 算法　Viterbi algorithm　113,203
Watson-Crick 互补　Watson-Crick complement　51,94

ISBN 7-5025-5649-4

9 787502 556495 >

ISBN 7-5025-5649-4/Q·104　定价：48.00元

销售分类建议：生物学/生物科学